World
RetroMagazine
future days are back
ISSN 3103-697X
Issue 1 Year 1 - June 2020 - https://www.retromagazine.net - Free Publishing

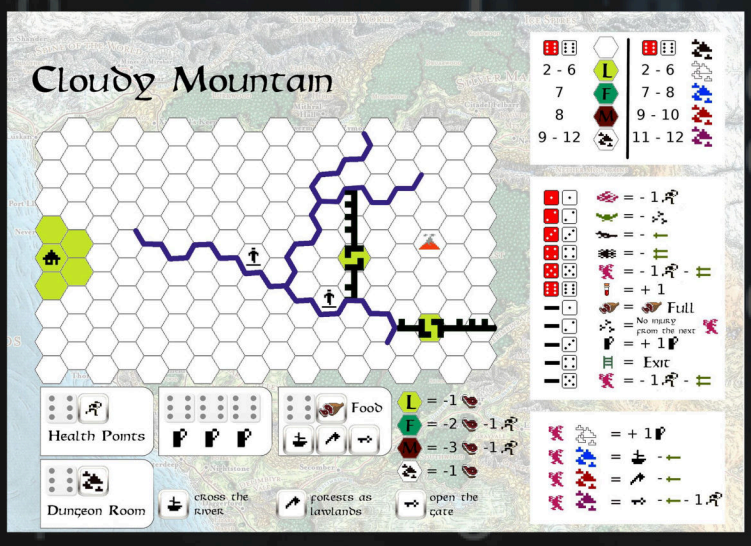
SONIC THE HEDGEHOG™
Sega Megadrive / Genesis review

- 3D Graphs with few lines in BASIC
- The corner of darkness - SEPTERRA CORE
- Playing infinite lives with your C64 - "The challenge goes on"
- MISTer FPGA, one year later...
- FAST Basic – a Locomotive Basic compiler in CP/M

RetroMath: secret codes

Interview with Francesco Sblendorio
Meeting the designer and developer of PETSCII-BBS

CHUBBY GRISTLE (C64 review)



CLOUDY MOUNTAIN - A single player boardgame

GTA: the origins of the myth

Spring 2020: there's a scent of change in the air

Despite the frankly somewhat dark times we live in, this late spring brings many changes in our lives. Perhaps even finally a return to life as we know it, after the CoViD-19 nightmare. Even within our editorial initiative there are no shortage of changes on the horizon. Indeed, many have already started or will soon be under way.

Let's start with the name of your (hopefully) beloved magazine. From this issue the name of the magazine changes to **RetroMagazine World**. We have been thrifty and modest: we have only added a small word ("World") to our historical name, mostly in order to show our new intention to address the entire international community and no longer only our numerous Italian readers.

How do we intend to do this? Well, actually, we already did it last May 2nd, with the release of issue zero of RetroMagazine English, a pilot publication entirely in English, dedicated to all the retrocomputing, retrogaming and retrocoding fans scattered all over the planet. These readers have long been asking us to bring in a "neutral" language (an official language, understandable to all) for the content and columns that for over two years have been reaching Italian readers. The success of this release has gone beyond all expectations and has definitively convinced us to continue the publications in English. So, starting from this issue #23, a complete English translation will be published about a week or two after the Italian one. The English release of RMW will also include a selection of the most interesting articles and reviews from the first 22 Italian issues.

Also our logo and general graphic design have been slightly modified, with the addition of a small planet in 8-bit graphics (see cover) and the appearance of a new slogan ("**Future days are back**"), which deserves a few words of explanation. Sure, it's a bit nostalgic, but it also tries to combine our past of being geeky kids and tireless gamers with the future that many of us saw then, imagining maybe even making a profession out of what at the beginning was just pure passion. At that time, we had the feeling that everything was possible in a world that was gradually opening up to the digital age and we felt a constant amazement when faced with games, applications and graphics/sound demos wonderfully generated by our humble 8-bit computers. In this present of ours, somewhat very uncertain, we can still rediscover those feelings, going back to the origins and to a simpler conception of computer science by using our old machines in a modern way. One of the best methods to achieve this outcome is by reading the issue you have in your hands right now. Because that's what we at RMW do: we try to give you back those days of light-hearted, optimistic fun, without forgetting the technological advantages that the present can provide.

The novelties for RMW and for all of our readers do not end there. Just a few days ago we opened our official Twitter profile **@RetroMagazineW**, which we invite you to follow to be always updated on our publications, read our posts with previews and news about the latest games and applications produced by the whole international community.

From this issue on, we are publishing our magazine through the **Creative Commons 4.0 International** license, reaffirming the absolutely non-profit nature of our editorial initiative, free of advertising and totally free for our readers. On the last page of this magazine you can find the "Disclaimer" box with the details about the distribution license.

Still not enough news? Then let us tell you that we are also working on the new RMW website and many other exciting retro-projects that we will reveal in the coming issues.

The issue you have just downloaded is full of interesting and hopefully stimulating content. Among the many great contents we'd like to highlight the article on **Cloudy Mountain**, an original and exciting board game, created by Giovanni Galli. This is another way to feel, at least for a few hours, that your future days are back! Happy reading to all!

David La Monaca

Email: retromagazine.redazione@gmail.com
Facebook: www.facebook.com/RetroMagazine-2005584959715273/
Twitter: twitter.com/RetroMagazineW

SUMMARY

◊ MISTER FPGA, one year later... Page 3
◊ Interview with Francesco Sblendorio Page 7
◊ FAST BASIC – a Locomotive Basic compiler in CP/M Page 14
◊ Star Watcher Page 17
◊ Playing infinite lives with your C64 – The challenge goes on Page 21
◊ Retromath: Secret Codes Page 24
◊ 3D Graphs with few lines in BASIC Page 27
◊ Japan cronicles: A new Game & Watch? Page 32
◊ How I discovered RPG games on my TI99/4A Page 36
◊ KNIGHTMARE SAGA (MSX) Page 42
◊ CITY BOMBER 4K (C64) Page 44
◊ CLOUDY MOUNTAIN – a boardgame for one player Page 45
◊ GRAND THEFT AUTO I (MS DOS) Page 48
◊ TRAKERS (Amstrad CPC) Page 50
◊ CRYSTAL CASTLES (MS DOS) Page 52
◊ CORNER OF DARKNESS SEPTERRA CORE (PC) Page 54
◊ SONIC THE HEDGEHOG (SEGA) Page 56
◊ ASTAL (SEGA) Page 58
◊ POWER SPIKES (Arcade) Page 60
◊ CHUBBY GRISTLE (C64) Page 62

People involved in the preparation of this issue

- Alberto Apostolo
- Carlo N.Del Mar Pirazzini
- Daniele Brahimi
- David La Monaca (Cercamon)
- Edoardo Ullo
- Ermanno Betori/ Alessandro Tiso
- Federico "Arzak1" Gori
- Flavio Soldani
- Francesco Fiorentini
- Giuseppe Fedele
- Marco Fiaschi
- Marco Pistorio
- Michele Ugolini
- Roberto Lari
- Starfox Mulder

More credits

- Graphic support: Irene G. Valeri
- Cover: Flavio Soldani
- Proof-reading: Francesco Fiorentini, David La Monaca, Robin Jubber, Giorgio Balestrieri, Alberto Apostolo, Leonardo Giordani, Gianluca Girelli

MISTER FPGA, one year later...

by Roberto Lari

Hi everyone, it's been a year since I last wrote about some FPGA systems (MISTer, MISTICA and MIST) on RetroMagazine (Italian version) and in this article we'll talk about the MISTer, which in my opinion represents the most powerful FPGA system in 2020. Together we will discover and examine all the updates and news of these last 12 months that made the MISTer even more interesting and irresistible.
If you are not familiar with the MISTer and FPGA systems in general, I invite you to look for information on the Web, as my previous articles appeared in issues #13, #14 and #15, which at the moment are only available in Italian. RetroMagazine World will publish these in the next few months. In short, these FPGA-based systems will introduce you to a whole new world of reproducing our beloved 8- and 16-bit vintage computers, as well as the old consoles, and even an ever increasing number of arcades (i.e. video games as they were available in bar cabinets then typically converted to home computers). FPGA systems have absolutely nothing to do with the typical software emulation that you may know as they run on your PCs and Raspberry PIs.
Let's dive into the MISTer features. The Linux operating system, running on the ARM CPU, is now equipped with a real shell (retrievable by pressing F9 on the initial menu screen). From there, after logging in with user "root" and default password "1", we can issue some of the basic commands of the operating system. It's actually a small environment and not a full Linux OS installation.
This "micro Linux" has absolutely no impact on the quality of FPGA hardware reproducing consoles or computers as it is completely independent and transparent.
From the Linux console you can launch the MC program (the classic Midnight Commander), a file management tool that allows 2 windows on the screen to easily create folders, copy or move files, rename them, delete them, etc. It is very comfortable because it also supports the use of a USB mouse and external mass storage devices (drives and/or hard disks) to exchange files to and from the MISTer MicroSD card.
But be careful, external memories cannot be controlled inside the computer/console cores, so you cannot use them to load games directly.
Speaking of Linux shells, if you have the MISTer connected to the Internet (which I highly recommend), you can also access it from your PC via SSH, establishing a connection through a program such as Putty (Windows). From every SSH client you can perform the same operations, such as running updates, modifying the configuration when necessary, etc.. From your Windows PC (if the home router allows it) you can access the MISTer MicroSD (also mapping it as a network drive) to easily add, move, remove and reorganize the files inside it. You can even mount a VHD file (virtual hard disk used by some cores such as Amiga, MSX, PC486, to name a few).
If you connect your MISTer to the Internet, you will also get the automatic update of time and date. This might be useful with some cores such as PC486 and Amiga. Even if your LAN router is not properly configured for SSH, you can always access via FTP using a client like the good old FileZilla.
Another MISTer interesting feature is booting from an external memory device. It works but I didn't test it thoroughly. What I have noticed is that from an external device you cannot launch scripts but this feature aside everything is quite the same as the with the usual MicroSD. And yes, it feels very comfortable when you can boot up from a multi-terabyte hard disk!
Each core you use has its own rules, but generally they look for games and ROMs within folders that have specifica names. If your games and applications are copied into these folders, it will be easier for the core to find them, as typically cores are not allowed to reach the root folder of the MicroSD. This of course forces us to navigate inside the subfolders in order to find the game file we are looking for, but generally the core searches directly into the folder with its corresponding name.
For example the Minimig AGA searches for the folder "Amiga", the C64 core searches for the folder named

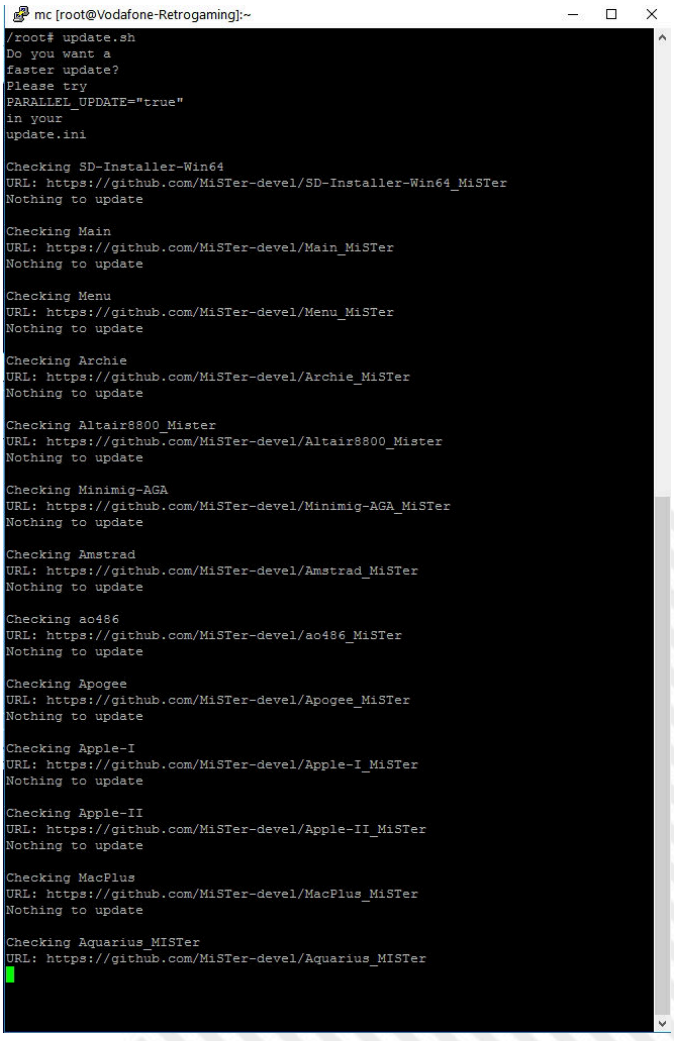


Figure 1 - Updating the OS via SSH

Figure 2 - A simple On/Off switch cable

“C64”. Please note that this doesn’t work for any core: the PC Engine/Turbo Grafx core actually searches for a folder named “TGFX16” and not something simpler like "PCENGINE”.
Here is a list of some of the default folders searched by the most popular cores available:
AMIGA, AMSTRAD, APPLE-II, AO486, AQUARIUS, ARCHIE, ASTROCADE, ATARI800, ATARI2600, ATARI5200, ATARIST, BBCMICRO, C16, C64, COLECOVISION, GAMEBOY, GENESIS, MACPLUS, NES, ODISSEY2, PET2001, QL, SAMCOUPE, SMS, SNES, SPECTRUM, TGFX16, TI-99_4A, VECTREX, VIC20, ZX81
One thing left me pleasantly surprised: the management of 2 different screens at the same time (HDMI and CRT RGB TV via VGA cable > SCART). You can run any core simultaneously on both the video outputs and so compare images, colours and anything else about image quality.
Another feature tha I found out by chance is that the MISTer cores can load games inside ZIP compressed files, of course in read-only mode. But I didn't go any further, no ZIP files for me.
Which one of you doesn't know about ScummVM? That’s the software that allows you to play the classic graphical adventures such as Monkey Island 1 & 2, Zak McKracken, Loom, Maniac Mansion and many others. Well, there is a version that works pretty good for the MISTer!
The software can be installed manually using a script (it is much easier to do than to explain the procedure). The script simply takes advantage of the Linux environment and the presence of the ARM chip. ScummVM is a real "must" for fans of this genre, playing Sam'n'Max or Full Throttle is really a great experience!). The MISTer currently supports version 2.2 of this engine and it also can run Monkey Island 3, which was released for Windows 9x takes up to 2 CDs. To fully enjoy the ScummVM engine, I suggest you to set the resolution on the HDMI screen to 640x480 as if it were a DOS PC in VGA, so the interface and games will use the widest possible portion of the screen (4:3 and black vertical bands on both sides).
PAL and NTSC resolutions for FPGA cores have their own independent settings (when I use a CRT monitor via RGB SCART I set it with 720x576 PAL 50 Hz and 720x480 NTSC 60Hz respectively, but if I used a single HDMI screen for everything, I would have set it to 1280x720 at 50 and 60 Hz). When using the Linux shell, you can install text adventures (such as the Zork Trilogy), through a special script called Interactive Fiction.
Over the past year, every core (or almost every core) has been updated and new ones have been added to the list, including Chip8, HT1080Z (a clone of the TRS-80 model I), AY-3-8500 "Pong-on-a-chip", Oric-1 and Oric Atmos, Arduboy and finally a good hardware reproduction of Atari ST. Talking about updates of the cores, the most noteworthy one is definitely Minimig AGA (Amiga): in the latest release they have optimized it so much that they have obtained a great increase in speed. It is so high (in the most extreme configuration) that it has become 33% faster than a real Amiga 1200 with an ACA1233n accelerator board with a 68030 CPU clocked at 40 MHz and 128MB RAM. The developers also have greatly improved the reproduction of the 68020 CPU by adding many missing instructions, making it much more compatible with a greater number of games in AGA mode. Minimig AGA now supports up to 384MB of the MISTer’s DDR3 RAM memory availbale on the DE10 Nano motherboard. This memory is configured as FAST RAM by the Amiga core, whereas a total of 2MB CHIP RAM is configured from the 32/128 MB of MISTer’s SDRAM. I have tried running Gloom Deluxe on this core and it was smoother than ever!
Why 128MB of SDRAM, you must be wondering... Yes, a MISTer version with 128MB SDRAM was also put on the market. I find it particularly useful to have 128MB for some NEOGEO games and little more, but if you are thinking about purchasing this system, the 128MB version asks for a little difference in price if compared to the 32MB version. I suggest you to buy the 128MB version and you won’t regret it, because it will surely support all of the upgrades for its lifetime.
We mentioned the NEOGEO because its official core has been out for a few months now. The completely free of

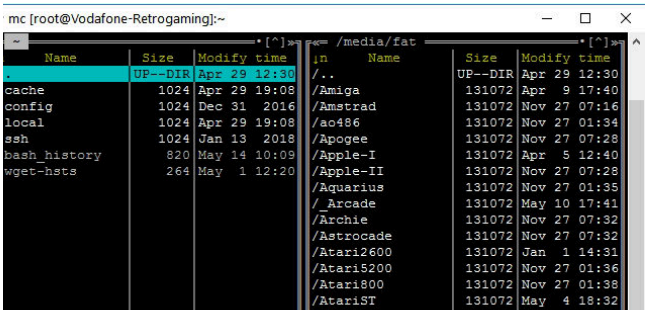


Figure 3 - MC running on Linux via Putty/SSH

charge Universe BIOS version 4.0 is perfectly compatible with the NEOGEO core.
As for the SEGA Megadrive fans out there, the support for CDs has been added to the core, thus creating the MegaCD core, with full support for CUE/BIN files, actually expanding the gamebase available for this platform. At the time of writing, I got an email announcing that shortly the core of the PC Engine CD will also see the light! So, as you can see, the MISTer is a constantly evolving system these days.
With regards to the Arcade cores, the number of games supported is constantly growing. Several games from the Sega System 1 platform (Flicky and Wonderboy, to name a few) have been rebuilt and another set is also coming for the Capcom Play System 1 platform. Possibly by the time you are reading this article, these arcade games

could already have become available.
When running the C64 core you can now autostart a CRT cartridge such as the Action Replay 6.0 (equipped with Fast Load, which is my favourite even over the Jiffy-DOS) by copying its .CRT file inside the C64 folder and calling it “boot3.rom”. Doing so, every time we start the C64 core, all of the functions of the cartridge will be available at once. Very comfortable, don't you think?
Still talking about the C64, in the most recent versions of the core, support has been added to the T64 format files and to a second drive (ID: 9), with an on/off switch (some demos and games do not work if they detect a 9 drive on). A really useful function especially with programs like GEOS, where you use two drives at the same time, one for the applications and the OS and one for your own documents. In addition, recently the support for the VIC-II chip has been considerably improved, thus allowing to solve several glitches in some games and many demos. The improvement is a really remarkable and noteworthy one so that the overall working of the VIC-II can be easily defined as the almost perfect reproduction!
Unfortunately, not everything is perfect. There are still some bugs and many things to solve. But hey, you can’t have anything, can you? Here follows a list of the key-points to be improved (well, point 2 is easily solved, point 3 we can only hope for an improvement over time and point 1 nothing can be done or so):
1) Vertical Arcade cores: the 90 degree rotation needed to play arcade games without having to physically rotate the screen vertically is only possible on the HDMI output. On VGA/RGB monitors it behaves exactly like the MIST and MISTICA (the horizontal screen-based arcades have no problem).
2) The MISTer does not have an ON/OFF power switch, but that can be easily solved by buying a few centimetres long cable with an ON/OFF switch. It only costs a few euros/dollars.
3) The PC486 core is slightly comparable in performance to a 486SX 25 MHz CPU based PC, perfect for modest games like Aladdin but insufficient if you have to settle for 3D games like Doom. We hope for future improvements as happened to the Minimig AGA (Amiga) core.

Here are other information about the MISTer, some of them have been reported to me by friends and FB users : The fan switch on the MISTer is not used to turn it on and off but only to adjust its speed to high and low speed. It’s not possible to turn it off completely, but setting it to low speed almost cancels the noise.

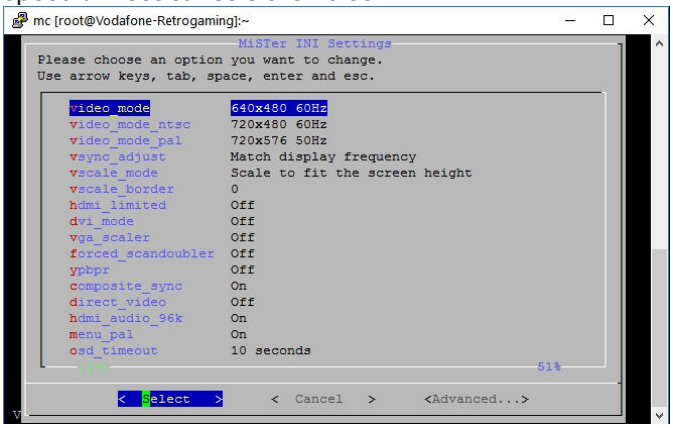


Figura 4 - Reviewing settings via SSH

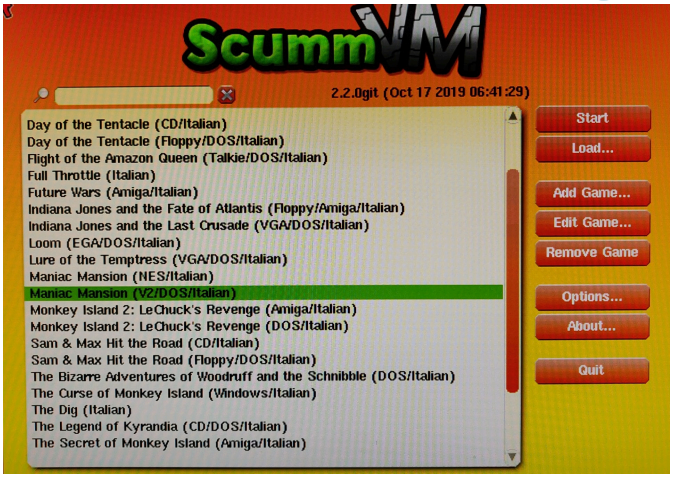

ScummVM
2.2.0git (Oct 17 2019 06:41:29)
Day of the Tentacle (CD/Italian)
Day of the Tentacle (Floppy/DOS/Italian)
Flight of the Amazon Queen (Talkie/DOS/Italian)
Full Throttle (Italian)
Future Wars (Amiga/Italian)
Indiana Jones and the Fate of Atlantis (Floppy/Amiga/Italian)
Indiana Jones and the Last Crusade (VGA/DOS/Italian)
Loom (EGA/DOS/Italian)
Lure of the Temptress (VGA/DOS/Italian)
Maniac Mansion (NES/Italian)
Maniac Mansion (V2/DOS/Italian)
Monkey Island 2: LeChuck's Revenge (Amiga/Italian)
Monkey Island 2: LeChuck's Revenge (DOS/Italian)
Sam & Max Hit the Road (CD/Italian)
Sam & Max Hit the Road (Floppy/DOS/Italian)
The Bizarre Adventures of Woodruff and the Schnibble (DOS/Italian)
The Curse of Monkey Island (Windows/Italian)
The Dig (Italian)
The Legend of Kyrandia (CD/DOS/Italian)
The Secret of Monkey Island (Amiga/Italian)
Start
Load...
Add Game...
Edit Game...
Remove Game
Options...
About...
Quit

Figure 5 - ScummVM 2.2

The 8GB MicroSD that comes with the MISTer purchase is not provided by the seller, but comes together with the DE10 Nano motherboard. Basically there is no difference for those who buy the complete and ready to use MISTer. Nothing much, but I wanted to clarify.
As you can imagine, the MISTer is a constantly developing platform. In a year it has changed a great deal and who knows what else we will have available in a year.
If you are wondering to purchase and FPGA-based system and choosing between the MISTer and the other devices available such as the MISTICA or Mystery, I would like to share my thoughts and experience to help you take the right decision.
The MISTer is priced about 300 Euro, if you opt out for the 128MB SDRAM version, an RGB SCART video cable included and, as mentioned before, an ON/OFF switch cable (see the link below). In my opinion it is worth the price. Why? Because you can have dozens of computers, consoles and arcades available in one single device. You can play games at will, with RGB quality for all compatible cores, with a portion of space in your desk and you can even use the same TV/monitor screen for all the machines/cores. In addition, think about how much would each of these machines cost if you had to buy them one by one (assuming you could find them for sale and all in working condition).
Just consider, for example, the Apple IIe with basic accessories or the Amiga 1200 with an accelerating board such as the ACA1233n board with a 68030/40MHz CPU and 128MB of RAM memory. Talking about the original consoles (Megadrive, Super Nintendo, PC Engine, etc.), if you wanted to use them with the same comfort (games on a SD card), you would have to equip each physical machine with an Everdrive card, so you would have to spend a lot more money.
Another example? The NeoGeo AES console: how much does it cost? Probably the console alone would overcome the cost of the MISTer (not to mention the cost of each game or the purchase of an Everdrive card).
In conclusion, if somebody asked me today about saving 120/130 euros and going for a different FPGA system, I would say no. Better go for a MISTer. The purchase would be one and only and you would get the best product available now and for a long time yet. No doubt about it. There are already other FPGA-based system similar to the MISTer (MIST and MISTICA) but today they lack the

chance to reproduce some of the high-end platforms (NeoGeo, PC486 and some others). They probably will not support many oter cores that require more RAM and more CPU power. Unconfirmed rumours refer that a Playstation 1 core is already underway.
Another big difference, as mentioned before, is about the support of the vertical arcade cores. You can play them on the MISTer, even though you get black bands on both sides of the screen using the HDMI output only. On the other systems you have to physically rotate the screen. In addition, as seen before, the MISTer can be accessed via Windows network, SSH and FTP, while the other system can only be fed with MicroSD cards.

The MISTer may also use EXFAT as its file system, so a single file can be as large as 16GB. On the other systems the only supported format is FAT32 with a maximum file size of 3.99GB. MISTer can run the ScummVM engine and it features 7 USB ports instead of just 4 provided on MISTICA and MIST. The possibility to use both video outputs at once is also an exclusive MISTer feature.
Last but not least, looking to the future, it is likely that more and more FPGA core developers will be able to move to the MISTer platform, abandoning the previous systems. If so, you may find yourself with a product that despite being valid, may no longer receive updates, not to mention the fact that some cores are not the same or compatible yet, since the different FPGA capacity oamong the systems. For example, the Minimig AGA core (Amiga) does not support more than 24 MB of FAST RAM on MIST and MISTICA and the maximum achievable performance is half the one you can get on the MISTer. The MIST/MISTICA C64 core does not support the second 1541 drive on ID 9. All these difference basically depend on the FPGA chip, which is 4.5 times larger on the MISTer and this of course concurs to let the developers implement many more features of the machine being reproduced.
The configuration of USB joysticks and gamepads is as simple and immediate on the MISTer as it is more cumbersome on other systems. The reason is that on the MISTer the support comes directly from the OSD itself, whereas on the other systems you have to edit the MIST.INI file, often you have to to go for trial and errors before finding the right setting.
I'd like to state that I still think that MISTICA and MIST are both excellent products and that the cores made for these systems are still working fine and very faithful to the machines they try to reproduce.
But, in my opinion, the point is that if I didn't own any of the 2 other FPGA systems today and I had to buy one, frankly I'go for the MISTer because I can't find any good reason not to.
If you are now rushing to Manuel Fernández's website to place your order for a MISTer, remember that it is not currently available. You will need to contact him via email and book your system in a queue. Manuel always receives a lot of bookings and typically it takes up to 3 weeks from the booking to his message that confirms that your MISTer is ready to ship.

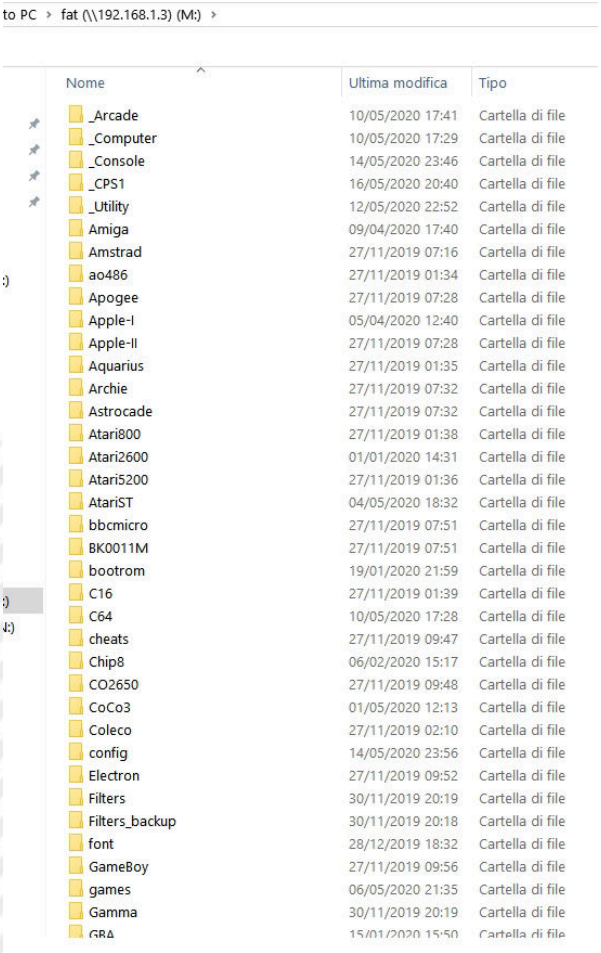

to PC › fat (\\192.168.1.3) (M:) ›
Nome
Ultima modifica
Tipo
_Arcade 10/05/2020 17:41 Cartella di file
_Computer 10/05/2020 17:29 Cartella di file
_Console 14/05/2020 23:46 Cartella di file
_CPS1 16/05/2020 20:40 Cartella di file
_Utility 12/05/2020 22:52 Cartella di file
Amiga 09/04/2020 17:40 Cartella di file
Amstrad 27/11/2019 07:16 Cartella di file
ao486 27/11/2019 01:34 Cartella di file
Apogee 27/11/2019 07:28 Cartella di file
Apple-I 05/04/2020 12:40 Cartella di file
Apple-II 27/11/2019 07:28 Cartella di file
Aquarius 27/11/2019 01:35 Cartella di file
Archie 27/11/2019 07:32 Cartella di file
Astrocade 27/11/2019 07:32 Cartella di file
Atari800 27/11/2019 01:38 Cartella di file
Atari2600 01/01/2020 14:31 Cartella di file
Atari5200 27/11/2019 01:36 Cartella di file
AtariST 04/05/2020 18:32 Cartella di file
bbcmicro 27/11/2019 07:51 Cartella di file
BK0011M 27/11/2019 07:51 Cartella di file
bootrom 19/01/2020 21:59 Cartella di file
C16 27/11/2019 01:39 Cartella di file
C64 10/05/2020 17:28 Cartella di file
cheats 27/11/2019 09:47 Cartella di file
Chip8 06/02/2020 15:17 Cartella di file
CO2650 27/11/2019 09:48 Cartella di file
CoCo3 01/05/2020 12:13 Cartella di file
Coleco 27/11/2019 02:10 Cartella di file
config 14/05/2020 23:56 Cartella di file
Electron 27/11/2019 09:52 Cartella di file
Filters 30/11/2019 20:19 Cartella di file
Filters_backup 30/11/2019 20:18 Cartella di file
font 28/12/2019 18:32 Cartella di file
GameBoy 27/11/2019 09:56 Cartella di file
games 06/05/2020 21:35 Cartella di file
Gamma 30/11/2019 20:19 Cartella di file
GBA 15/01/2020 15:50 Cartella di file

Figure 6 - Browse MicroSD content through LAN

Useful links:

• https://manuferhi.com/c/mister-fpga
Developer and seller of Mister

• http://www.atari-forum.com/viewforum.php?f=117
Mister: Help and Discussion forum

• https://tinyurl.com/ycjapgwf
Italian Group totally devoted to FPGA Mister, Mistica & Mist

• https://tinyurl.com/y54rmza3
International Group totally devoated to Mister

• https://github.com/MiSTer-devel
Github with cores update (dialy updated)

Thanks to:

The author would like to thank the following friends, who made this article about MISTer a reality.

Thanks for your collaboration to:
Angelo Caltabiano; Rino Rinaldi; Mauro Sanna; Matteo Alvares; Paolo Barbieri; Wladimiro Porcelli; Carlo Piacentini; Massimiliano Marongiu; Cristiano Rosadini; Alessandro Bertolotti; Danilo Savioni; Marco Galliani; Massimiliano Conte; Fortunato Pavone; Tommaso Tautonico; Vincenzo Scarpa; Moreno Ligabue.

Interview with Francesco Sblendorio

Meeting the designer and developer of PETSCII-BBS

by David La Monaca (Cercamon)

Here on the pages of RetroMagazine World we keep on meeting developers, technicians, programmers and active participants of the international retrocomputing and retrogaming scene. This time we asked some questions to Francesco Sblendorio, a Senior Java Engineer, home computer and retrocoding fan and creator of PETSCII-BBS, a Java open source solution to quickly activate a software server that acts as a BBS and is able to support terminals based on the Commodore variant of the ASCII character encoding. The PETSCII-BBS framework, whose sources Francesco published on GitHub, serves precisely to put on a real BBS (Bulletin Board System), just like those of the past 80s and 90s, which were connected via modem and a simple PSTN telephone line. Today many BBSes are still active and accessible via the Internet. A BBS built upon PETSCII-BBS is no exception. On the client side it only requires a terminal program that supports the PETSCII encoding or has been written specifically for the C64 (40 columns) and the C128 (80 columns).

Figure 1 - Francesco Sblendorio

Intro

Let's first get to know Francesco who, with this project, surprised the whole community of Commodore fans. He also got a very successful feedback during the presentation of his work at the last conference “Once Upon A Sprite” held last October 2019 in Milan, Italy.

Hello Francesco and welcome to the pages of RetroMagazine World. We at RMW were attending your OUAS 2019 conference and, like the rest of the audience, appreciated the simplicity and the effectiveness of your work on PETSCII-BBS and so, we thought we would transfer these feelings to our readers. But before we talk about your raids in the world of 8-bit programming, let's get to know you a little bit better. How about we talk about your personal history, your educational training and your profession?

Hi all, I was born in 1977 in Turin, grew up in Bari and then moved to Milan since 2005. I attended science high school and then graduated in computer science, a subject for which I had a passion since the early years of school. Other passions I have are music (strumming guitar and ukulele), contemporary history and science. I volunteer for some critical thinking outreach associations. My job is to develop professional software solutions.

Now a ritual question for a magazine like ours: what was your first computer and what do you remember about it?

My first computer was a Commodore 16, a mysterious object that came into my house when I was eight. Initially I had a certain distrust of that machine (these are the memories of a child discovering something completely new), then my interest started increasing. In particular I remember a programming course released on cassette called “Video BASIC”, published by the Italian Jackson Editorial Group, which I used to follow release after release. My inner desire was to get to the end of the cassette where there was one (and sometimes even two) video games. I remember I used to seethose games as a “reward” after studying the BASIC lessons.

With the help of my father, who randomly bought magazines and books with the inscription "Commodore” on them, I slowly grew that passion that today is my job. At one point the C16 became a more powerful and popular C64 and I continued cultivating my curiosity for programming.
I used to browse the pages of Commodore Computer Club (A/N: a famous Italian magazine of the time) initially without understanding much, then browsing them over and over again. I also used to mechanically copying the BASIC listings published and slowly beginning to understand the operations involved, finally replicating them in programs that I tried to write myself.

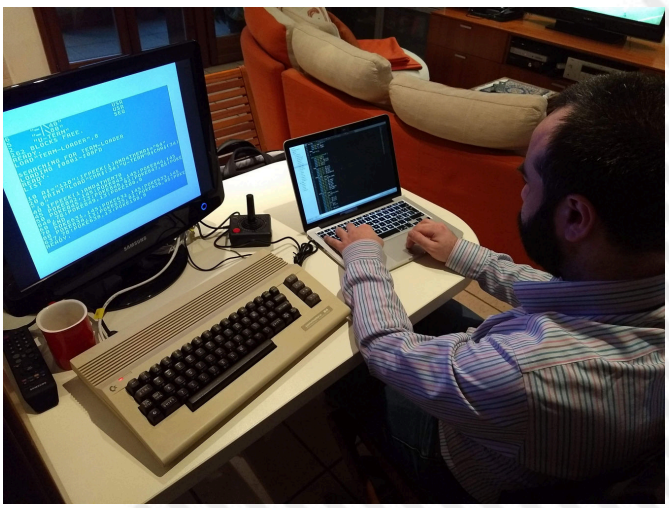


Figure 2 - During a coding session

Figure 3 - "Commodore Therapy": relax during test phase

"Papersoft" (A/N: a cheap Italian magazine of the 80s including BASIC game listings) is also part of this plethora of memories. I remember that this magazine contained programs for many platforms (TI-99/4A, Spectrum, Apple II) that I did not own, but whose characteristics I could tell by reading the published listings. So I used to spend my time copying those games and translating them to the C64 version of BASIC. I mean, not exactly a hobby for a kid, but I enjoyed it!
Becoming a computer geek this way, the passion for retrocomputing is always part of me as it traces back to my origins and, why not?, it also brings me back with nostalgia to my youth.

Another mandatory question for you: what was the first game you remember loading up to your first home computer?

I remember it very well, it was "Munch It", a Pac-Man clone for the Commodore 16. I was always looking forward to coming home from school to play that video game.

When did you find out about the world of programming and realise that you could create something with a computer instead of just playing games and apps?

Almost immediately, as a result of reading, first randomly and then systematically, computer magazines, which came into my hands a little "randomly" thanks to my dad.

How did you learn to program? Like many of us in the 80s and 90s, were you self-taught or did you attend specific courses or studies?

I was basically self-taught until I was 18 years old, although at some point I also attended a course where I could get my hands on PCs with MS-DOS, they were inaccessible to me differently. Then I undertook university studies that gave shape and substance to the bubble that was in my head before!

Can you tell us the name of the 8/16-bit system that has made you more passionate than anyone else and that you think is your favourite? And which is the game that kept you glued to the screen the longest?

Certainly Amiga, which made me go from 8 to 16 bits and to programming with languages such as Pascal and C. The Amiga almost ironically also allowed me to practice the world of MS-DOS through the very slow AmigaTransformer emulator.

Talking about programming, did you start with BASIC, like most of us, and then move on to assembly or other high-level languages?

I started with BASIC and approached Assembly 6502 very superficially, then moved on to high-level languages like Pascal, C, then C++, Java, Python, etc. Today I look back with historical interest at languages like Modula-2 and Forth.

When did the moment occur (sadly, in a way) when, like everyone else, you had to switch to purchasing a PC with its x86 architecture, for work or passion?

I switched from my beloved Amiga 500 to a 486DX2 PC in 1994 but I don't remember it as a sad moment. On the contrary, it was a significant leap in quality: I went from the Amiga 7MHz 68000 CPU with 1MB RAM and no hard disk to a "big" 66MHz 80486DX, 4MB RAM and 200MB hard disk. It was not for work because I was a student, but just for my university studies. The MS-DOS operating system together with MS Windows 3.11 gave me a lot of satisfaction. My nostalgic side still led me to search for and use emulators such as CCS64 and UAE and I gladly remember the first successful attempts to dump the C64 and Amiga floppy diskettes and then spin them on the emulators.

Speaking of work and passion, what are the retro computers and systems that you use today? Are you a collector, too?

I am not a collector, I am more interested in being able to own the retro computers that I can use mainly for study purposes. For example, I currently own two C128, but mostly for "backup" reasons. I'm not interested in things like owning that particular model or keeping intact seals or collecting "low" serial numbers. I used the C128 to study the CP/M operating system and how it could be programmed. On the RetroAcademy website there are several articles that I have written on this topic and on my GitHub repository [http://github.com/sblendorio] there are some projects in Pascal, C and Modula-2 all exclusively written for CP/M and in particular for the Commodore 128.
Another example: I used a Commodore MPS-803 printer (paid 30 euros) to write a small driver that makes it usable to print text and graphics (monochrome) from a modern computer running Windows, Linux or macOS (these projects can also be found on my GitHub).
A couple of years ago, thanks to the devices produced by Pasquale De Luna (the KC64Eth cartridge and the KC64Wifi modem) and to Gideon Zweijtzer's 1541U2+ cartridge, I "discovered" the networked Commodore 64 and that's the topic that I am most passionate about at the moment. Now I went from the original C64 to its Ultimate64 reproduction, a motherboard also "made in Gideon Zweijtzer". The world of FPGA replicas intrigues me a lot,

because it makes our beloved vintage machine somehow "eternal".

I think I already know the answer, but I'm still going to ask you the question. Do you like technical solutions and modern peripherals applied to old computers like the C64?

As you can imagine, I love them! Just when I found out about the existence of the SD2IEC interfaces in 2013 I was enthusiastic about it. Indeed, it is perhaps just when I read about the use of these interfaces as mass peripherals or the existence of MIST and FPGA technology that I "picked up" my old passion for vintage computers.

In addition to the Commodore models you mentioned so far, which other 8-bit and 16-bit computers in the past tickled your interest (both in games, architecture, etc.)?

Undoubtedly the Sinclair ZX Spectrum, because of its user interface, so very particular or, rather, unique. In general I would like to get my hands on CP/M machines from the late 70s or early 80s, like the KayPro models, when CP/M was at its peak.

When did you (re)start programming on retro systems? What was the trigger that led you on the road of retrocoding?

In 2001 I found the newsgroup it.comp.retrocomputing (where I met people involved in retroprogramming, such as Elia Bellussi of MuPIn, Daniele Lena and Davide Bressanini of Brusaporto Retrocomputing, and also Enrico Barbisan who at the time used to organise Spresiano Retrocomputing) and the pleasant memory of the computers with which I started my adventure in computer science came back on. So I started looking for the machines I was fond of and the ones I had always wanted but never had the opportunity to put my hands on. As you can imagine, after a short time the room in my house began to run out and that available to computers gradually became a sort of a "warehouse". I just put them aside and I didn't have any desktop space to actually use them, so I quit. After a few years I moved to Milan and through Facebook I rediscovered passion through the groups, and so I decided to reserve a desk to two or three systems and to dedicate myself to them, writing code and actively using them.

Figure 4 - RetroAcademy BBS: main screen

What are the main programming tools you use for your retrocomputing projects?

Mainly the CC65 compiler (C language) to produce object code for Commodore 64 (I was intrigued by this compiler reading two articles by Fabrizio Lodi) but I also use server-side Java (in order to produce code THAT DOES NOT run on retrocomputers). Java is mostly used to create application servcers to which vintage machines can connect, as in the case of running a BBS that can be created with PETSCII-BBS. Recently, the announcement of the "10liners competition" made my friend Fabrizio Caruso repeatedly urging me to participate with listings in BASIC and I believe that this time he will prevail and I will have a go at it too.

Let's now talk about the PETSCII-BBS project, the Java framework that allows anyone to activate their own BBS on the Internet for C64 and C128 terminals. How did you come up with that idea?

Well, not only for C64 and C128, but also for C16 and Plus/4 and theoretically also for the PET (the key point is to have a way to connect them to the Internet, for example via the user port). The idea actually was born as a joke: when I worked in JobRapido (an Italian job search engine service), a colleague who knew my passion for the retro computers jokingly told me "Would you be able to write a version of JobRapido that works on the Commodore 64?". And this is my colleague a week later (see figure).

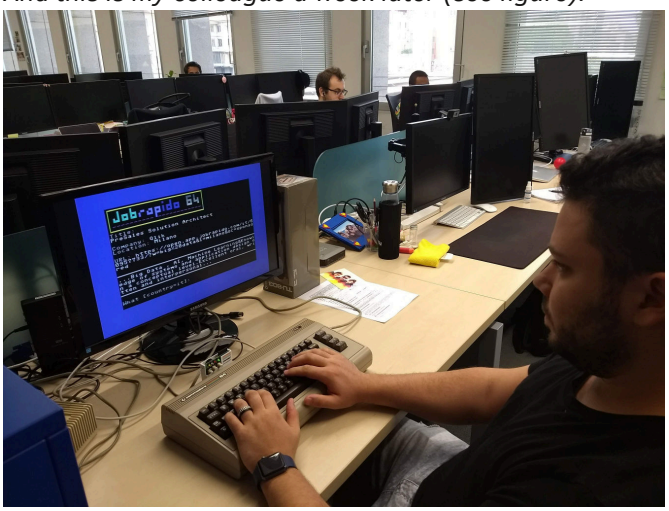


Figure 5 - One of Francesco's colleagues browsing JobRapido on a C64

I had just purchased Pasquale De Luna's KC64Eth card to connect the C64 to the Internet via the RR-NET standard protocol and had tried for the first time to connect to a BBS with KipperTerm. My colleague's joking request made me wonder: "How do I implement a BBS-like service, programmable to my liking, that works with a C64?" So I found around the Internet several software to create a BBS, but they all offer the same functions: software downloads, message boards and more, all customizable with PETSCII graphics, but all in all nothing "open source" and most of all nothing that could be easily changed at the will of a developer.
Therefore I thought about writing a simple framework that, masking the management of the connection between

Figure 6 - BBS screen on C128 and C64

server and client (using sockets, error management, protection from possible "spurious" accesses) only shows input and output functions to and from the terminal to which you are connected, along with tools to handle the PETSCII encoding. The result was PETSCII BBS Builder.

Why did you choose Java to build the server side of your application?

Because it is simply the language that I use for work and which I have a solid competence on. As a result, it was very easy for me to implement the "core" functionality, thus linked to multithreading and the management of TCP sockets. You could also implement the same functionality in Python, C#, or other languages.

What were your overall design and deployment choices for the server platform?

I started from the idea of writing a "core" that implements and "masks" the TCP connection, the management of the multithread and the conversion of the PETSCII charset, so as to provide the user-programmer with a kind of DSL that can be used to design the functions without knowing the details masked by the core, and that's how it was.
The project mainly provides two packages: a package

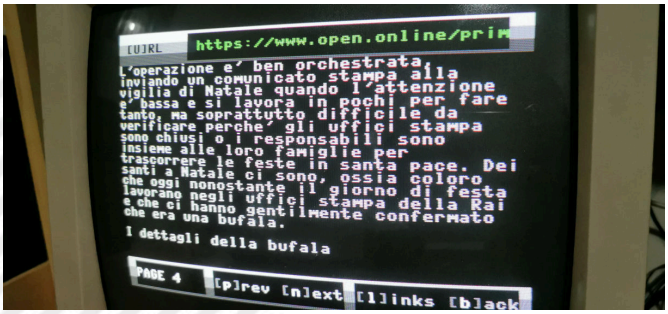

[U]RL https://www.open.online/prim
L'operazione e' ben orchestrata
inviando un comunicato stampa alla
vigilia di Natale quando l'attenzione
e' bassa e si lavora in pochi per fare
tanto, ma soprattutto difficile da
verificare perche' gli uffici stampa
sono chiusi e i responsabili sono
insieme alle loro famiglie per
trascorrere le feste in santa pace. Dei
santi a Natale ci sono, ossia coloro
che oggi nonostante il giorno di festa
lavorano negli uffici stampa della Rai
e che ci hanno gentilmente confermato
che era una bufala.
I dettagli della bufala
PAGE 4
[p]rev [n]ext [l]inks [b]ack

Figure 7 - RetroAcademy BBS: web browser

called "core" and another called "tenants", where you can find numerous examples that use the core, ranging from games such as Connect-4 and 15 Puzzle, to WordPress JSON feed parsers, to messaging services, etc.
What I called "Retroacademy BBS" is an online and active project, available at "bbs.sblendorio.eu" or "bbs.retroacademy.it", port 6510. It was born as an example of what can be achieved with my framework. In fact, the purpose of the project is to provide the core and push potential user-programmers to design the services.

How difficult is it for PETSCII-BBS users to create their own BBS online once the server part is installed?

It's not complicated at all. You must first install the JAR package on the server and then implement your own class representing the BBS (extending the PetsciiThread base class) [A/N: refer to box 1].
There is a simple example provided with the framework that actually creates a real BBS with a few lines of code and uses only the I/O functions provided by PETSCII-BBS. In practice, it only does very simple things: it prints a welcome string and asks the visitor for the username, then greets the visitor by repeating the name entered by the user. The example, deliberately simple, only serves to show the supplied Domain Specific Language (DSL), which allows you to abstract from network functionality, focusing in this case only on input, output and processing.
Among the demos already present in the project are examples of increasing complexity and there is even an implementation of the historic Zork text adventure (thanks

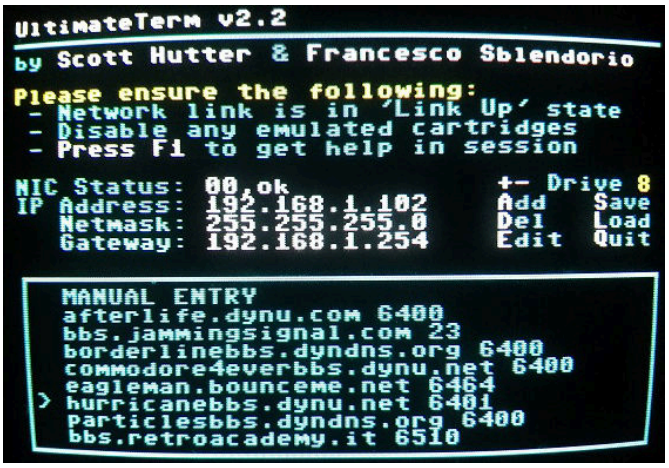


Figure 8 - Ultimate Term in action

to Roberto Manicardi, using Wei-ju Wu's ZMPP project).

We now come to the development of the client terminals and the many utilities featured by the D64 image "Ultimate Term and Demos". Can you tell us briefly about these tools?

Of course I do. I had just started my experiments with the core of the PETSCII BBS Builder on the newborn "BBS Retroacademy" and had tested it with the KC64Eth cartridge that implements the RR-NET protocol. I already had bought the 1541U2+ cartridge which includes an Ethernet port, but it could not be used with the terminal emulation programs that make use of RR-NET.
Shortly thereafter, my friend Federico of Dato informed me that he had found a new program called "Ultimate Term", posted on the Facebook group "1541 Ultimate" dedicated to the 1541U2+ and the Ultimate64 board. "Ultimate Term" was the work of Scott Hutter, but my tests systematically crashed when he connected to BBS Retroacademy.

I was pleased that users of the 1541U2+ and/or Ultimate

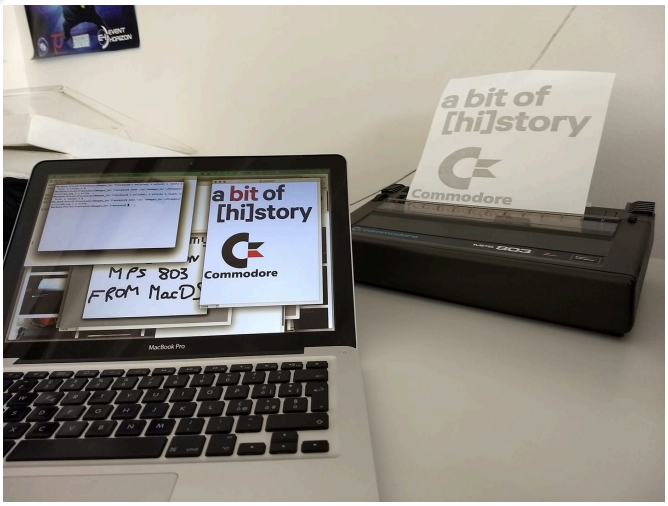


Figure 9 - Print of graphics from PC on a MPS803

64 could potentially connect to many BBSes and thus also to those produced with my framework, so I downloaded the program to watch it crash and understand why. I contacted Scott Hutter, who immediately pointed out that the program was open source and that the project was freely available and could be downloaded from GitHub (https://github.com/xlar54/ultimateii-dos-lib).
In fact, the project, a bit like mine for the BBS, was aimed to expose a core, a library (in this case in C for the Commodore 64) able to offer the programmer in a transparent way the network functionality of the 1541U2+/ U64. It also provided as an example a draft terminal emulator, which was in fact Ultimate Term, provided together with the library.

Unlike the library, which was carefully documented in every detail, the examples including Ultimate Term were still in an initial state (they were just samples). Examining the code of Ultimate Term (version 1.3) I noticed a couple of bugs: one was the buffer overflow that crashed it when it connected to Retroacademy BBS, the other was a "rendering" problem, which created glitch and slowed the output.

So I started fixing these bugs and I really liked the result. Then Scott Hutter made me co-author of the repository that since then has been maintained and updated by me. It was still an embryonic example, it allowed nothing more than connecting to a BBS without being able to download files or maintain a list of BBS instances except in hard-coded mode and it was still optimizable in terms of I/O speed on the screen.
Slowly and with the advice of Antonio Savona, I further optimized the program and equipped it with bookmark management, the ability to download using the XModem protocol and send DOS commands to the disk drive. The development is currently at a standstill, but I would like to provide the software with additional features, such as implementing XModem upload functionality and adding Punter as a protocol for both download and upload.

We know that PETSCII-BBS can interface with many connection systems that offer C64 and C128 a link to the Internet network. Can you tell us which are the main ones and, in your opinion, which are the best in terms of price and performance?

I know three:
1. Via the Wifi modem. There are many on the net, very similar to each other. Personally I use and recommend KC64Wifi by Pasquale De Luna. It reaches 9600 bps when used with CCGMS software (https://csdb.dk/release/?id=156523). The cost of the modem is 23 euros shipped, this is the most economical solution. Speed is, among the three solutions proposed, the lowest, but for navigating BBS it is not a problem at all.

2. Using a compatible RR-NET cartridge. In this case you connect via the Ethernet port of the cartridge, for example using KipperTerm (https://csdb.dk/release/?id=174537). The cost of the cartridge goes from 40 to 60 euros. The speed reaches 10Mbps, considerably higher than the Wifi modem. For BBS navigation, the screen refresh rate is noticeably higher.

3. Using the 1541U2+ cartridge or the Ultimate64. Until yesterday, you had to use UltimateTerm (https://csdb.dk/release/?id=181558) or CCGMS Ultimate (https://csdb.dk/release/?id=174485). I didn't measure the speed but it's comparable to the RR-NET cartridge. With the latest firmware update, however, it is possible to activate modem emulation through Ethernet, so CCGMS can be used in a "classic" way, setting a 19200 bps Swiftlink modem as a connecting device. The cost in this case is 150 euros (for the 1541u2+ cartridge to be connected to the C64) or about 200 euros for the entire Ultimate64 board (which replaces an entire C64). It is the most expensive option but obviously includes all the other countless advantages of the products themselves!

Is it possible to connect to a BBS built with PETSCII-BBS using a terminal client program that runs on a suitably configured VICE emulator?

Some people did it, I didn't! In fact, if anyone wants to write a guide on this, that would be very welcome. Connecting to PETSCII BBS from PC is essential for me to test the Builder and I found a simpler way, using two alternative programs:

1. SyncTerm - https://sourceforge.net/projects/syncterm/ available for Mac, Windows and Linux

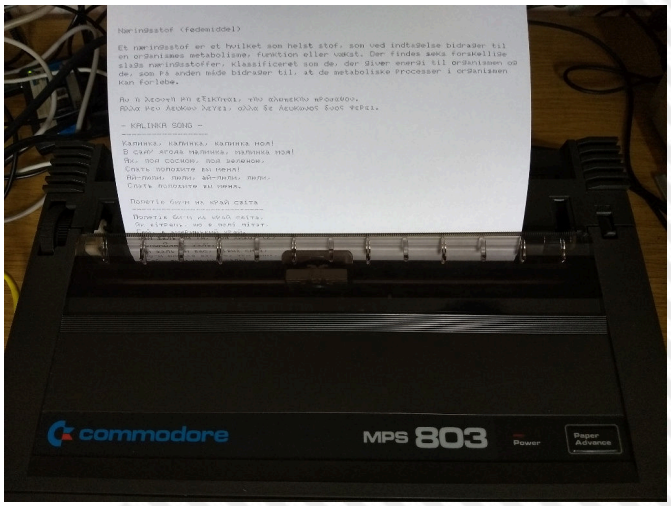


Figure 10 - Print of text on a Commodore MPS803

```
public class WelcomeBBS extends PetsciiThread {

    // NEVER forget default (empty) constructor
    public WelcomeBBS() {}

    @Override
    public void doLoop() throws Exception {

        // clear screen
        cls();

        println("This is your brand-new BBS");
        println();
        print("Enter your name: ");

        // flush output
        flush();

        // clear input buffer
        resetInput();

        String name = readLine();
        println();
        println("Welcome, " + name + "!");
        println("Press a key to exit");
        flush();
        readKey();

    }
}
```

Box 1 - sample code to create a very simple BBS

2. CGTerm - https://csdb.dk/search/?search=cgterm
Here is a video made by a friend Dani Hibari to connect via SyncTerm - https://www.youtube.com/watch?v=2FbMa9Ragr0

We mentioned "RetroAcademy BBS", a board that currently offers many services (news from many web sites, online games, chat services and even a text-only web browser). Do you know of other BBS around the world that have been activated thanks to your framework? What feedback have you received from so many C64/128 enthusiasts since you made your work public?

The "Retroacademy BBS" was born as a demo, as a "case study" included in the project to only show what can be done with the framework. It also shows "how to". Despite its nature of "demo", however, it is getting a fairly good success. I have not compiled usage statistics, but I often see posts sharing information on PETSCII-BBS on many Facebook groups. They often include C64/C128 and sometimes even Plus/4 screenshots connected to this BBS. The most successful applications are reading RAI Televideo (the Italian official teletext service) and downloading files from the CSDb website through the BBS. Also the internal chat and the text-based web browser are the most interesting services. At the time of writing, there are at least 2 BBSes using my framework:
The Old Net - theoldnet.com:6400
Temporary Vortex BBS - bbs.c64.xyz:6400
There is also a Retro + Halloween Night video on YouTube where my framework was used to produce a "fake BBS fake": https://www.youtube.com/watch?v=l6i0e3JrV7A

Speaking of RetroAcademy BBS, did you do it all by yourself or did you ask for and get the help and support of other developers? What do you foresee in the future of this BBS, now a reference point for all 64 users equipped with a network interface?

There are several people who have contributed and whom I thank for the current state of Retroacademy BBS:
Piero Corasaniti, who contributed the version of the CSDB-Download that allows the download of "whole" D64 files, intended for download to SD2IEC.
Roberto Manicardi, who helped interface Infocom games (Zork is the most famous) with the BBS, extending ZMPP (an open source interpreter of Infocom games)
Wei-ju Wu, the author of ZMPP (http://zmpp.sourceforge.net/)
Richard Bettridge a.k.a. "ssshake", who contributed to the text browser developed for his BBS "theoldnet" (https://github.com/ssshake/theoldnet-bbs)
Felice Pagano, who helped me optimize the code.
Henrik Wetterström, for his project "droid64" (http://droid64.sourceforge.net/) that I used in the download section from CSDb.
Jatin Thakur for the AI engine I used in the Connect-4 game (https://github.com/jn1772/Connect4AI)
Brian "bigbhowell" for the AI engine I used in the "Tic-Tac-Toe" game (https://github.com/bigbhowell/tic-tac-toe).
The artists who provided their works in the "PETSCII art gallery" section (Shine, Markku Reunanen, John Canady, Manuel Vio, Ivan Mavilla, Davide Fichera).

For the future, after implementing the chat, and having therefore figured out how to manage multiple users, who eventually select each other in a session, I plan to create some online multiplayer games. Even something as simple as the classic battleships, but with a remote human player: I think it's fun and educational!

Being open source, I hope that the framework will be used by other users, such as those I mentioned earlier (namely, the BBS "oldnet" and "vortex"), with whom to exchange ideas and new code. For example, the framework currently lacks the functionality of "receive" with regard to the XModem protocol and the Punter protocol management (both "send" and "receive") is totally missing. If anyone is willing to lend a hand, it's definitely welcome!

Do you expect to be able to open up the possibility of connecting to the BBS created with your framework to other back platforms? Even those that do not support PETSCII encoding for obvious reasons?

I gave priority to the PETSCII encoding because I became passionate about the Commodore 64 application. I think that in the future I will also adapt the framework to the ANSI/VT-100 standard so that I can use terminal emulators on PC and Amiga. It's not complicated, but I'd like someone to help me. I still appeal to those who listen/read me!

What other projects related to the world of retrocomputing are you currently working on or planning to work on shortly? Are there any other home computer platforms that attracted your interest in addition to the 8-bit Commodore platforms?

As I said, I really like the CP/M world, so I made four games

in Pascal and Modula-2:
Gorilla (in Modula-2) https://github.com/sblendorio/gorilla-cpm, Queens (in Turbo Pascal 3) https://github.com/sblendorio/queens-cpm, Hanoi (in Turbo Pascal 3) https://github.com/sblendorio/hanoi-cpm, Hangman (in C) https://github.com/sblendorio/hangman-cpm

In particular, the first and last (Gorilla and Hangman) use a library that I called "xterm" ("cross-term") that is available in both C and Modula-2. In Modula-2: https://github.com/sblendorio/mod-xterm-cpm, in C: https://github.com/sblendorio/c-xterm-cpm

This library allows you to use screen management functions (clear screen, cursor positioning, changing colours, etc.) regardless of the type of terminal used (for example, the C128 uses the ADM-31 standard, the Amstrad CPC uses Zenith Z19). Two other projects concern the Commodore MPS 803 printer, for which I wrote two "drivers" to be used in conjunction with the "opencbm" package, which allow printing with this UTF-8 plain-text printer and monochrome graphics by connecting it to a normal PC or Mac (with the "cable" XUM1541): cbmage, for images: https://github.com/sblendorio/cbmage; cbmtext, for UTF-8 text files: https://github.com/sblendorio/cbmtext

Outro

We kindly thank Francesco for his time and for the spirit of openness he has shown regarding his projects. As we have witnessed in person, PETSCII-BBS and its main implementation "RetroAcademy BBS", they do work great and continue to attract a lot of interest from many C64/C128 fans.

The sensation of using real hardware dating back to the early 1980s connecting to the Internet and accessing really useful and efficient services is insane, especially for those who, like myself and many of our passionate readers, were born and grew up with a "limited" home computer like the C64. The PETSCII-BBS framework and its implementations in the form of BBS and online services are a typical example of how Commodore's small/large 8-bit is still booming in a thousand useful projects and solutions thanks to the experience, creativity and imagination of developers and technicians of yesterday and today.

The dedication and the passion shown by Francesco are precisely what we are trying to bring to the attention of readers on the pages of RMW, shedding light on ideas and projects that combine solid foundations and modern personal information technologies. Just for fun!

Links and bibliography reference

• Project PETSCII BBS Builder on GitHub: https://github.com/sblendorio/petscii-bbs

• Game "Gorilla" in Modula-2 for CP/M systems: https://github.com/sblendorio/gorilla-cpm

• Library "XTerm" (cross-terminal) in Modula-2 for CP/M systems: https://github.com/sblendorio/mod-xterm-cpm

• Library "XTerm" (cross-terminal) in C for CP/M systems: https://github.com/sblendorio/c-xterm-cpm

• Game "Queens" in Turbo Pascal for C128 in CP/M mode: https://github.com/sblendorio/queens-cpm

• CBMage: print bitmaps with a Commodore MPS 803 from PC or Mac: https://github.com/sblendorio/cbmage

• CBMtext: print text UTF-8 with a Commodore MPS 803 from PC or Mac: https://github.com/sblendorio/cbmtext

• Ultimate Term 2.3 for C64 e C128, by Scott Hutter & Francesco Sblendorio: https://csdb.dk/release/?id=181558

• Source of Ultimate Term: https://github.com/xlar54/ultimateii-dos-lib

• Card KC64Wifi by Pasquale De Luna: http://www.codingkoala.com/kc64wifi/

• Card KC64Eth by Pasquale De Luna: http://www.codingkoala.com/kc64eth/

• Coding in C on a Commodore 64 part 1, by Fabrizio Lodi: https://www.retroacademy.it/2018/10/23/retrocomputers/sviluppare-in-c-per-il-commodore-64-parte-1/6902/

• Coding in C on a Commodore 64 part 2, of Fabrizio Lodi: https://www.retroacademy.it/2018/12/01/retrocomputers/sviluppare-in-c-per-il-commodore-64-parte-2/6933/

• KipperTerm: https://csdb.dk/release/index.php?id=174537

• Specifications of RR-NET: http://wiki.icomp.de/wiki/RR-Net

• Project droid64: http://droid64.sourceforge.net/

• Project ZMPP (Z-Machine Preservation Project): http://zmpp.sourceforge.net/

FAST Basic – a Locomotive Basic compiler in CP/M

by Francesco Fiorentini

We already know Locomotive Basic as a versatile and rather fast language, but as it is an interpreted language, its speed is limited by the need to translate the instructions into LM at the time of execution. Now, assuming that we have already learned how to write code in Locomotive Basic, we may feel the need to speed up our work or simply create an executable file to distribute our programs. How can we do it?

We can switch to compiled languages... and sooner or later we'll do it, or we can compile our code with a Basic compiler.

In this regard, we tested a compiler released in 1988 by DMV Software (a German company) that responds to the name of Fast Basic Compiler. The peculiarity of this software is to run under CP/M 2.2 or CP/M 3.0. But how do we run this program on the Amstrad CPC? Follow us and we'll show you.

Before we start, however, let's make the shopping list of what we need.

1. Fast Basic Compiler. We can download it from here: https://www.cpc-power.com/index.php?page=detail&onglet=notices&num=4705,
at the same address you can also find a manual in German (for those who are familiar with this language) and a short, but convenient, guide in English (much more within my grasp).

2. CP/M 2.2. We can download it from here: http://www.cpcwiki.eu/index.php/CP/M_2.2

3. A disc with the programs in BASIC.

4. A real Amstrad CPC or an emulator like WinAPE or Caprice. I don't have a physical CPC, so I will use WinAPE emulator: download here: http://winape.net/

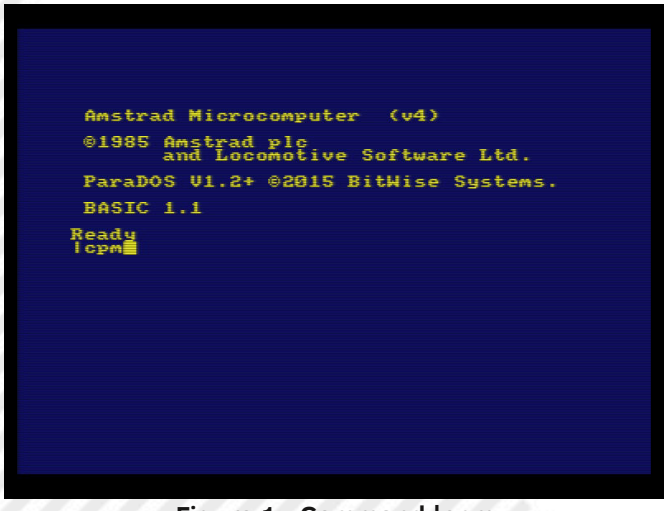


Figure 1 - Command |cpm

Figure 2 - CP/M 2.2

At this point we have everything we need to get started, but before we get down to business, let me say a few words about CP/M.

CP/M is a Digital Research operating system for micro and personal computer, developed in 1974 for the Intel 8080 processor but also compatible with Zilog Z80 processors. In the second half of the 1970s and the first half of the 1980s it had become a de facto standard in the industry with thousands of programs available. The Amstrad CPC, which mounts a Z80 processor, provided a bundled CP/M 2.2 disk drive to expand the software available to the user.

Now, after the quick introduction, let's move on to action.

Load the CP/M Operative System
Insert the CP/M 2.2 disk into drive A and type the command |cpm (Fig. 1).
After a brief execution time we should receive the prompt of the CP/M 2.2 Operating System (Fig. 2).
Let's launch THE DIR command to display the contents

CP/M 2.2 - Amstrad Consumer Electronics plc
A>dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRLKEYS COM : FWRESET COM : SUBMIT COM : RITDEMO BIN
A: ROINITIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOUCPM COM : PIP COM : SETUP COM
A: STAT COM : SYSGEN COM : XSUB COM : DISC BAS
A>

Figure 3 - Content of CP/M 2.2 disk

CP/M 2.2 - Amstrad Consumer Electronics plc
A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRLKEYS COM : FWRESET COM : SUBMIT COM : RIIDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOUCPM COM : PIP COM : SETUP COM
A: STAT COM : SYSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEM01 BAS : DEM02 BAS
A: DEM03 BAS : DEM04 BAS : DEM05 BAS : DEM06 BAS
A: DEM061 BAS : DEM062 BAS : DEM063 BAS : DISCMONI BIN
A)

Figure 4 - Content of Fast Basic disk

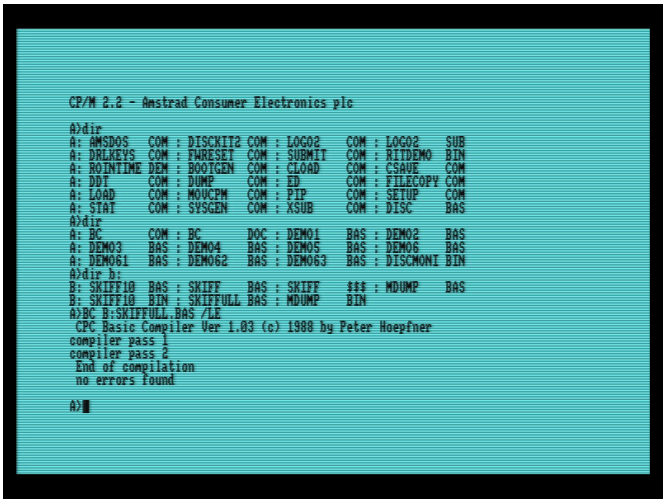

CP/M 2.2 - Amstrad Consumer Electronics plc
A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRLKEYS COM : FWRESET COM : SUBMIT COM : RIIDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOUCPM COM : PIP COM : SETUP COM
A: STAT COM : SYSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEM01 BAS : DEM02 BAS
A: DEM03 BAS : DEM04 BAS : DEM05 BAS : DEM06 BAS
A: DEM061 BAS : DEM062 BAS : DEM063 BAS : DISCMONI BIN
A)dir b:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$: MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)

Figure 5 - Successfully compiled!

of the disk inserted in drive A, as indicated by the command prompt: A > (Fig. 3).
Load the Fast Basic and the disk with the Basic programs
Now that the CP/M environment and its intrinsic commands are in memory, we can remove that disk and load the Fast Basic disk into drive A.
Once that's done, let's run THE DIR command again to see the contents of the Fast Basic.
You should see something similar to Fig. 4.
Now let's insert our disk with the programs in Basic in drive B and see its contents with DIR B:. Obviously the contents of your disc may be significantly different from mine.

Compile a program
Launch the command "BC B:SKIFFULL.BAS /LE" which means: compile the file SKIFFULL.BAS located on disk B: (ignore the/LE parameters for the moment).
Of course you will need to replace SKIFFULL.BAS with the name of your Basic program.

Once run, if your program does not have routines that the compiler cannot compile, you should see something similar to Fig. 5.

Just to confirm our program has been transformed into an executable (BIN), let's check the contents of disk B: again (Fig. 6).
As you can see, in addition to the SKIFFULL.BAS program, there is also the SKIFFULL.BIN executable.

Back to Locomotive Basic
Insert the CP/M disk back into drive A and launch the AMSDOS command (Fig. 7).

Now remove the disk containing the Basic programs from drive B, insert it into drive A and use the CAT (catalog) command to see its contents (Fig. 8).

Launch the compiled program
Let's launch our executable with the command run "SKIFFULL.BIN and enjoy the speed of the compiled game (Fig. 9 and Fig. 10).

Unfortunately, I can't show you the speed, but trust me when I tell you that it's about twice as fast as the interpreted version, so much fast that it's really hard to play it now.

A few technical information...
The compiler works on all Amstrad CPC 464, 664 and 6128 models with at least one disk drive.

The syntax, as we have seen, is as follows:

BC <filename> /L
BC <filename> /E
BC <filename> /LE

A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRLKEYS COM : FWRESET COM : SUBMIT COM : RIIDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOUCPM COM : PIP COM : SETUP COM
A: STAT COM : SYSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEM01 BAS : DEM02 BAS
A: DEM03 BAS : DEM04 BAS : DEM05 BAS : DEM06 BAS
A: DEM061 BAS : DEM062 BAS : DEM063 BAS : DISCMONI BIN
A)dir b:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$: MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)b:
B)dir
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$: MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
B)

Figure 6 - Here is the compiled .BIN

A: DEM03 BAS : DEM04 BAS : DEM05 BAS : DEM06 BAS
A: DEM061 BAS : DEM062 BAS : DEM063 BAS : DISCMONI BIN
A)dir b:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$: MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)b:
B)dir
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$: MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN : SKIFFULL BIN
B)a:
A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRLKEYS COM : FWRESET COM : SUBMIT COM : RIIDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOUCPM COM : PIP COM : SETUP COM
A: STAT COM : SYSGEN COM : XSUB COM : DISC BAS
A)amsdos

Figure 7 - AMSDOS Command

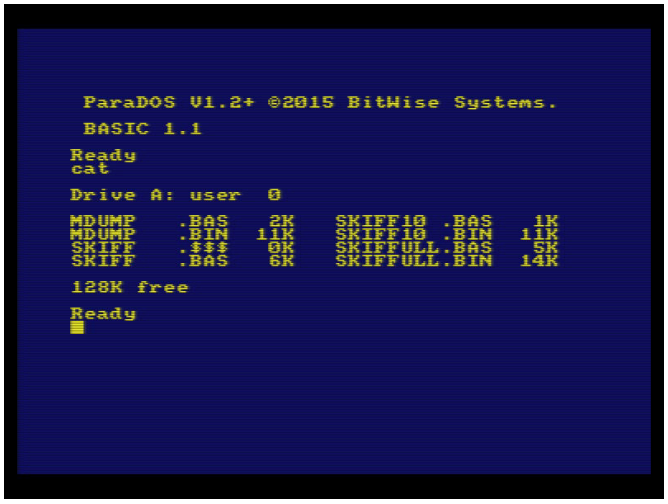


Figure 8 - CAT command on AMSDOS

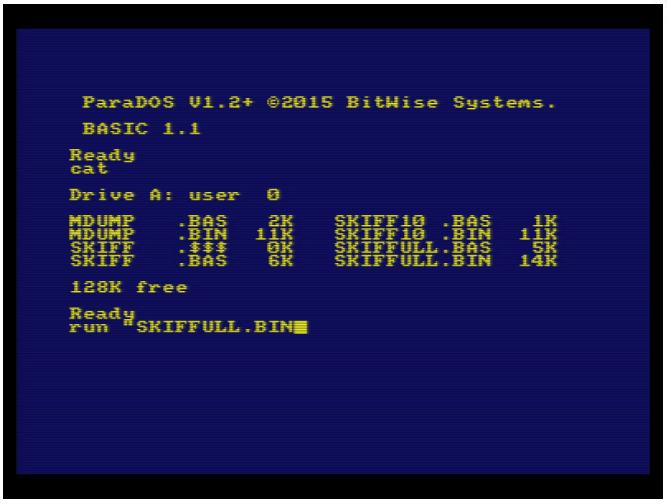


Figure 9 - Execute the .BIN file

Where the parameters have this meaning:

L – omits the line number in the compiled program (to save space)
E - to be used if interrupts are not needed (AFTER, EVERY, ON SQ)

All standard Locomotive Basic commands are supported, except for the following:

AUTO, CHAIN, CHAIN MERGE, CONT, DELETE, EDIT, LIST, LOAD (BASIC program), MERGE, RENUM, RESUME NEXT, SAVE (BASIC program), TROFF, TRON

Locomotive Basic 1.1 extensions are also supported!

Warning: it seems that the maximum size of compilable programs is around 17KB. I couldn't prove it, so take this information as just a warning.

If you want to speed up your programs further, you can optimize the compiler's performance by preferring INTEGER variables to floating-point variables (where possible, of course) in Basic code.

Floating point:

```
10 FOR i=0 TO 10000
20 NEXT I
```

Not compiled 11.1s, compiled 8.2s

Integer:

```
10 FOR i%=0 TO 10000
20 NEXT i%
```

Not compiled 5.9s, compiled 1.1s

You have no more excuses...

I look forward to receive your compiled programs!

Figure 10 - SKIFF in all its glory!!! :-)

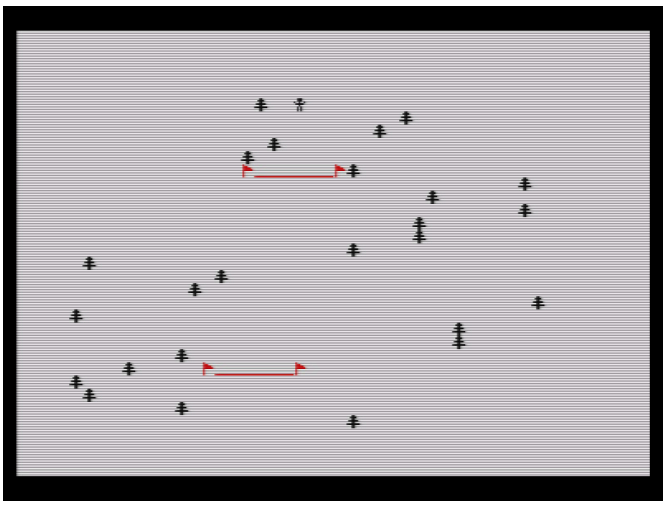


Figure 11 - WOW it's superfast!

Star Watcher

by Francesco Fiorentini

Star Watcher - Amstrad CPC

Software and manual can be download from:
https://www.cpcwiki.eu/index.php/Star_Watcher

In the last Italian issue (number 22) we show Visible Solar System, a space exploration program released in 1983 by Commodore for the C64. In this issue it is the turn of an astronomical program for Amstrad CPC released by AMSoft in 1984.

AMSoft was a software production company directly controlled by Amstrad and created specifically for the purpose of producing software for the CPC line. This approach contributed significantly to the success of the CPC, especially in the UK, France and Spain. We can therefore say that programs produced and released by Amsoft were official Amstrad software. The program we are reviewing is called Star Watcher and demonstrates the attention paid before releasing a package under the brand of AMSoft.

The manual

I have always appreciated the manuals accompanying the software, especially when they are well done and exhaustive. The Star Watcher manual is a clear example of the care and attention the software was released in the 1980s.

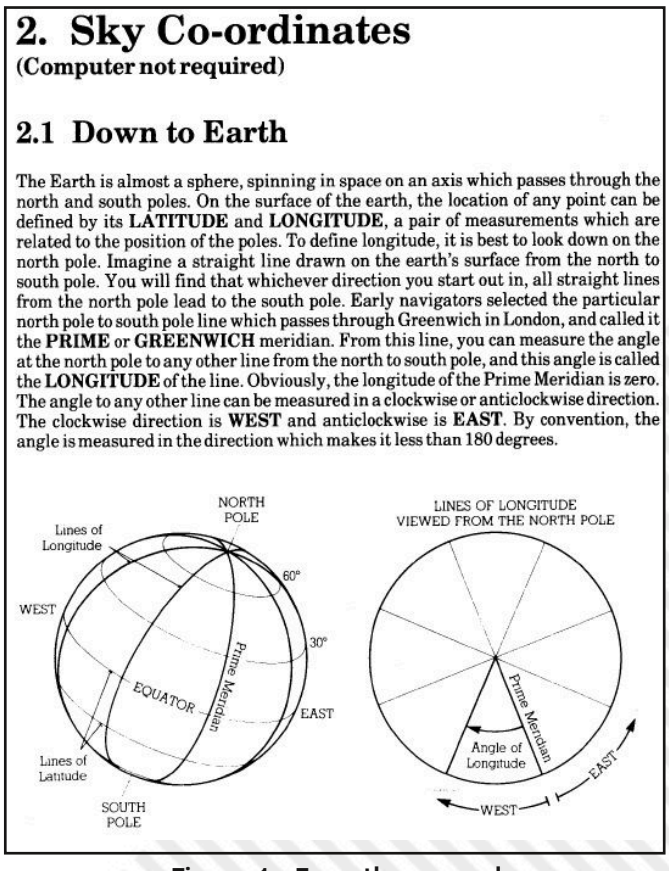
The manual consists of more than 50 pages containing detailed information about the software, a glossary of astronomical terms and as many as 3 appendices:
Appendix 1: constellation catalogue
Appendix 2: Trigonometric functions
Appendix 3: Software Load Instructions

But that's not enough, the software is structured in two distinct modules, the Star Watcher program itself and an educational module, designed to provide to the user a basic knowledge of astronomy in order to use the software at the best. In this respect, the manual plays a fundamental role:
the first 3 dispensations of the astronomy course are in fact in the manual and the computer is not necessary.

Let's see the course topics in detail:
1) explanation of how the sky is mapped and the conventions used to describe the position of stars (computer not needed)
2) how stars seem to move in relation to those observing them from Earth (computer not needed)
3) names of the stars (computer not required)
4) form and appearance in the sky of constellations (educational program required)
5) positional relation of the constellations (educational program required)

If, like me, you are not familiar with astronomical notions, you will find the manual very interesting and, just to learn something new, I advise you to get the software and read the manual carefully.

Before continuing, however, it is necessary to report some minimum notions to appreciate the Star Watcher program:
Latitude: is the angular distance between a parallel and the Equator. The latitude value is between 0° (Equator) and 90° (Pole), followed by the indication North (abbreviated N) or South (abbreviated S), depending on whether the location is in the northern hemisphere or in the southern hemisphere.
Longitude: is the angular distance between a meridian and the reference meridian, usually Greenwich. The value is expressed in degrees and followed by East (abbreviated E) or West (abbreviated W) depending on whether the measurement is made east or west of the meridian of Greenwich.
Magnitude: is the measure of the luminosity of a celestial object from an observation point (usually the Earth). Designed by Greek Hipparchus, it contemplated a scale from 1 (brightest star) to 6 (lightest star) and included only celestial objects visible to the naked eye. Today, thanks to modern technologies, the scale has been greatly expanded. The Hubble Space Telescope can observe objects up to the brightness value of 30.
Celestial Poles: North and South; these are the points in the sky immediately above the Earth's poles.
Celestial Equator: portion of the sky directly above the Equator.
RA Right Ascension: is the longitude projected onto the celestial sphere rather than the Earth's surface. Straight



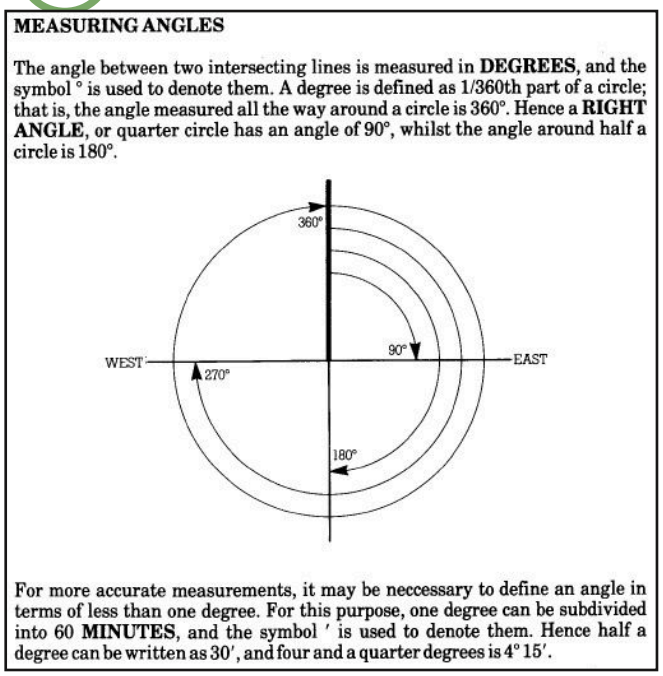
2. Sky Co-ordinates

(Computer not required)

2.1 Down to Earth

The Earth is almost a sphere, spinning in space on an axis which passes through the north and south poles. On the surface of the earth, the location of any point can be defined by its **LATITUDE** and **LONGITUDE**, a pair of measurements which are related to the position of the poles. To define longitude, it is best to look down on the north pole. Imagine a straight line drawn on the earth's surface from the north to south pole. You will find that whichever direction you start out in, all straight lines from the north pole lead to the south pole. Early navigators selected the particular north pole to south pole line which passes through Greenwich in London, and called it the **PRIME** or **GREENWICH** meridian. From this line, you can measure the angle at the north pole to any other line from the north to south pole, and this angle is called the **LONGITUDE** of the line. Obviously, the longitude of the Prime Meridian is zero. The angle to any other line can be measured in a clockwise or anticlockwise direction. The clockwise direction is **WEST** and anticlockwise is **EAST**. By convention, the angle is measured in the direction which makes it less than 180 degrees.

Figure 1 - From the manual



MEASURING ANGLES

The angle between two intersecting lines is measured in **DEGREES**, and the symbol ° is used to denote them. A degree is defined as 1/360th part of a circle; that is, the angle measured all the way around a circle is 360°. Hence a **RIGHT ANGLE**, or quarter circle has an angle of 90°, whilst the angle around half a circle is 180°.

For more accurate measurements, it may be neccessary to define an angle in terms of less than one degree. For this purpose, one degree can be subdivided into 60 **MINUTES**, and the symbol ′ is used to denote them. Hence half a degree can be written as 30′, and four and a quarter degrees is 4° 15′.

Figure 2 - Still from the manual...

ascension is measured in hours, minutes and seconds, corresponding to the Earth's rotation: 24 hours of straight ascension is a complete turn.
Dec Declination: is the latitude projected onto the celestial sphere rather than the Earth's surface. The points to the north of the celestial equator have a positive declination, those to the south, negative.
Az Azimuth: is the measure of the angle with respect to the North of the projection of an astral object on the horizon of the observer.

This information is all reported in the manual.

The Educational Module

By launching the command 'run “teach [enter]' you can start the educational module, a module, I remind you, that complement the first 3 chapters of the manual. After a splash screen reminding us that this software is part of a larger package called Brain Power, we will be greeted by a screen that requires to enter our latitude.

The reason is simple, in order to help us learn the name of the constellations and their location in the sky, the software, requesting our location, is able to draw the

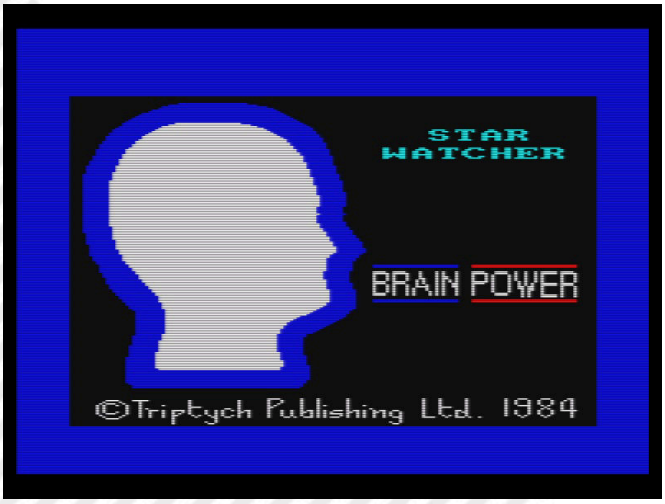


Figure 3 - A Brain Power's software

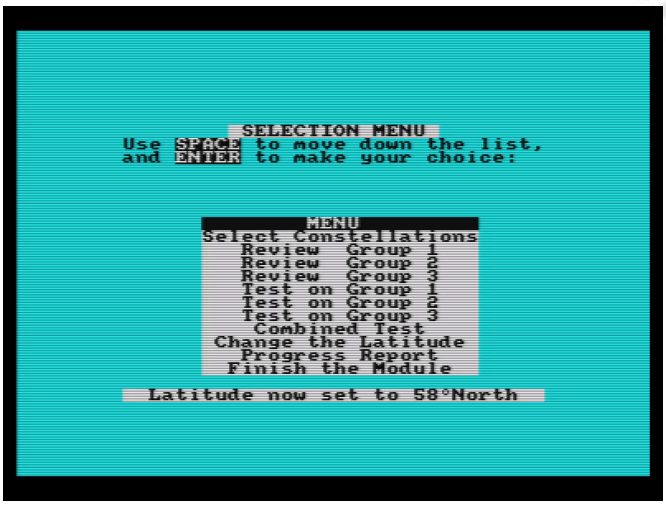


Figure 4 - Educational Module menu

constellations as they would appear above us, with many other surrounding stars.

The purpose of the educational program is to help us identify constellations by their shape (module 4) and by their shape and position in the sky with respect to other stars (module 5).

Figure 5 - Urs Minor without other stars

Figure 6 - Urs Minor with other stars

If the first module is relatively simple, the same cannot be said of the second. To help us on this learning path we can observe the constellations one at a time or divided into 3 groups.

Figure 7 - Our progress... (not so impressive...)

At the end of the learning path we will be able to test our ability through some tests. The results report is also interesting. Once we are satisfied with our educational path, we can close the module and move on to the actual program.

Star Watcher

By launching the command 'run “apply [enter]' you can start the Star Watcher program (...it's about time, you may say...) that will welcome us again with a request for information; this time more detailed than the educational module.

In fact, we will need to enter the following information:
- Today's date
- Current local time
- Local time at midday Greenwich (noon GMT)
- Our latitude
- Our longitude

Once this information is entered, we will finally be able to access the program.

Figue 8 - Here are our coordinates

We would expect to be greeted by a beautiful screen with a starry sky... but we would not!
The first screen is inexorably empty, but it already contains the entire control panel, divided into 6 areas to manage the program.

Figure 9 - Detail of the Main Screen

Here is the detail of the main screen (Fig. 9):
1) REAL TIME CLOCK: continuously displays the current time and date based on the data previously provided.
2) DATA DISPLAY: this is the area where the computer shows information about what is visible in THE SKY WINDOW and where you can define what to display.
3) SKY WINDOW: the window where the stars are shown. The portion of the sky is 90° wide and 45° high. The color of the window varies depending on the situation. At night the sky is black and the stars are white. By day the sky is blue and the stars black, to remind us that they are not visible.
4) MESSAGE LINE: the menu where to choose the actions. You can move between actions with the [SPACE] and confirm them with [ENTER].
5) STATUS DISPLAY: useful information for the user.
6) DIRECTION INDICATORS: divided into 2 parts, the upper part indicates the direction where we are looking, the lower part the angle in height from the horizon.

Let's draw the stars...

Well, at this point, I'd say we are ready to draw our first starry sky! As the manual suggests, let's look at a pre-set portion of the sky.
Use the spacebar to display the item 'Horizontal Plot' in (4) and then press [ENTER] to confirm. Press [ENTER] again at the request of the number of steps 'Number of Steps' and after a short wait you will see the screen as per figure 10.

Figure 10 - Horizon Plot

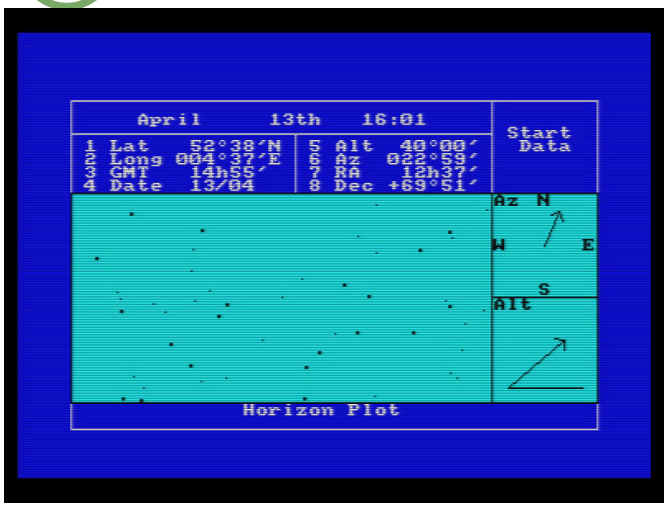


Figure 11 - 'Our' sky, blue because it's midday

Figure 13 - Magnitude selection

Let's draw 'our' stars...

Now we are ready to draw the sky above our position. We then scroll to find 'Starting Data' and after pressing [ENTER] we enter our parameters in (2). When we are satisfied with the data entered we will use the spacebar until the item 'Horizontal Plot' is displayed in (4) and after requesting the number of steps and a short wait we will see the screen as per figure 11. As anticipated, since it is midday locally, the sky is blue and the stars are black. Also try choosing the 'Complete View' option to increase the number of stars in our portion of the sky; figure 12.

Star magnitude selection

Now try selecting the 'Star Selection' option. After a brief loading the screen will be shown in fig. x. From here we

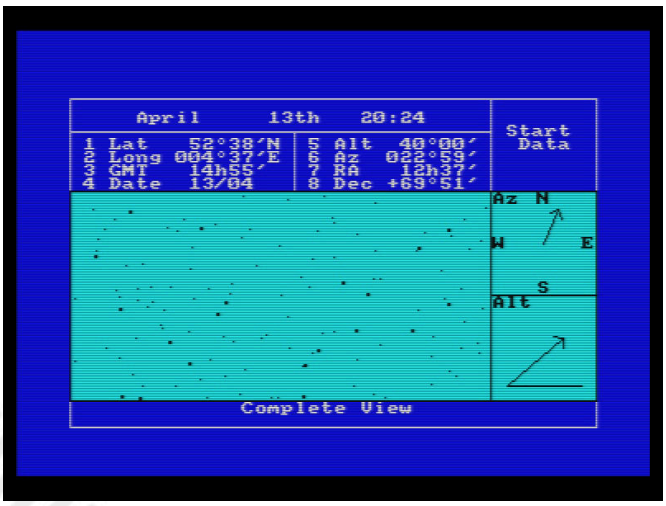


Figure 12 - 'Our' sky full of stars

can choose for each constellation which stars to display. Move with the cursor keys to the desired constellation and then press the numbers 0, 1 or 2 for the following options:

0) No star in the constellation will be drawn

1) Only stars that characterize the shape of the constellation will be drawn

2) All stars in the constellation will be drawn

We can also apply the choice to all constellations at the same time using the 'ALL' field.

Or choose a minimum magnitude using the field 'MAG' (Fig. 13).

Other Features

What has just been described is only a part of the numerous functionalities of the program that can be summarized in the following table (as also reported in the manual):

Complete View: Redesigns the current view with all constellations in mode 2 but subject to the currently set limit of magnitude.

Continouos Plot: draws the sky with an interval of 1 minute using the parameters provided in Data Display.

Copy to Printer: Sends the current screen to the printer

Cursor Mode: Activates a cursor and redesigns the screen by centering it at the cursor position once [ENTER] is pressed.

Equatorial Plot: Draws the sky with RA and Dec provided in Data Display.

Finishing Data: Edit Data Display for a final drawing in multi step view.

Horizon Plot: draws the sky with ALT and AZ provided in Data Display.

Real Time Plot:Draws the sky with an interval of 1 minute using the latitude and longitude initially provided.

Star Selection: allows you to choose which constellations to draw and a minimum level of magnitude to use.

Starting Data: Edits the dates in the Data Display.

Conclusions

The software is very quick to calculate the position to draw, find the stars in the database and draw them on the visible part of the screen. I am not able to measure its reliability, but I can highlight some shortcomings.

It is true that the program is called Star Watcher, but it would have been useful to include the planets in the solar system as well. Another thing I miss being an inexperienced user is the link between the various stars that form the constellations to define its shape. An option to turn it on/off in the main window would have helped a lot. Not to mention the lack of any information in the main window. Seeing the name of a constellation next to it would have helped the neophytes a lot.

Obviously, the software was intended for experienced astronomers or really convinced aspiring astronomers. The random user would find the software unconvincing.

Playing infinite lives with your C64 – The challenge goes on

by Marco Pistorio

Foreword

In this article many references will appear to "Infinite Lives with Commodore 64", which was published in issue 3 of RetroMagazine (only available in Italian at the moment). If you can't read Italian, no problem anyway, as the information and the instructions here provided can be used without any previous competence. The only prerequisites are a basic knowledge of the VICE emulator and some good will.

Introduction

After several years, I managed to play again a C64 game that in my youth I found very interesting. It is "PastFinder", which I knew about with a different title "Space supplies", as it was included in an unofficial tape called "Top Playgames" issue 13, which back in September 1986, in Italy, was distributed and sold by newsstands. Basically those tapes were pirate versions of the original games and this was "legal" in Italy only because of the lack of specific copyright laws about software programs and games.

The game stands out, in my opinion, for a discreet gameplay and the challenging game maps. It takes a steady hand and excellent reflexes to proceed through scenarios and levels that become increasingly hostile. Whenever you are able to complete a certain area of the game, you will discover a little more. At the time I could not go very far through the levels of the game, so I started to wish to see the whole game map in a different way: cheating! Completing the game was a task absolutely not easy without tricks or "trainers" made by some of the many active cracking groups.

Actually, I didn't consider that much the chance to find an already made cheat, because usually, if I may, I prefer to do the job myself. Today, I get the opportunity to tell you this little experience, also trying to provide you with the tools and the minimal knowledge to achieve similar results!

In a previous article, I introduced a technique to be able to play your favourite games with infinite lives. This technique works roughly in these terms:
a) Launch the game within VICE (the famous Commodore 8-bit machines emulator)
b) Take "snapshots" of RAM whenever you lose a life during the game
c) Examine these "snapshots" with a tool that helps you understand which memory location contains the information about the lives currently available
d) Modify the game code to prevent the number of lives contained in the identified memory location from decreasing.
In my previous article on the same subject, I've been using a program tool able to compare and review two or more different snapshots of the computer memory. More than two years have passed since that article and I rewrote the tool in Visual Basic.Net. You can download the tool along with its source code by using the links in the box at the end of this article.

Description of the activity

Let's get to work now and create two or three "snapshots" of the C64 RAM while the game is running. These snapshots will be created when 4 lives are available, then 3 and finally 2 lives only.
How can you create them? Quite simply, by opening THE VICE monitor and typing the following command string:

s "img1.mem" 0 0000 FFFF (Enter)

Please be careful and change the name of the file, from img1.mem to img2.mem, img3.mem and so on every time you lose a life in the game.
The instruction above is easily explained. Basically, we save the contents of the RAM between $0000 and $FFFF addresses in hexadecimal notation, which corresponds to the range 0-65535 in decimal notation, to our hard disk, in a binary file (img1.mem) of the file system (here is the meaning of 0 immediately after the file name).

At this point we can use the comparing tool. After copying these .mem files into the same folder as the tool, you can launch it. If you have carefully followed the instructions provided so far, you can set "img1.mem" as the file name in the first text box, which is intended to contain the name of the first file to be compared. Of course the second text box will contain: "img2.mem".

Finally, let's set a file name to contain the result of the comparison. I suggest a simple one: "out.txt".
Click "Start" to run the comparison and examine the result of the processing (see figure 1). The result tells us that there are several memory locations whose contents differ by 1. Our focus should be on 00FD.
Why is that? Because the first snapshot was taken when we had 4 lives, the second when we had 3. The two values therefore correspond perfectly.

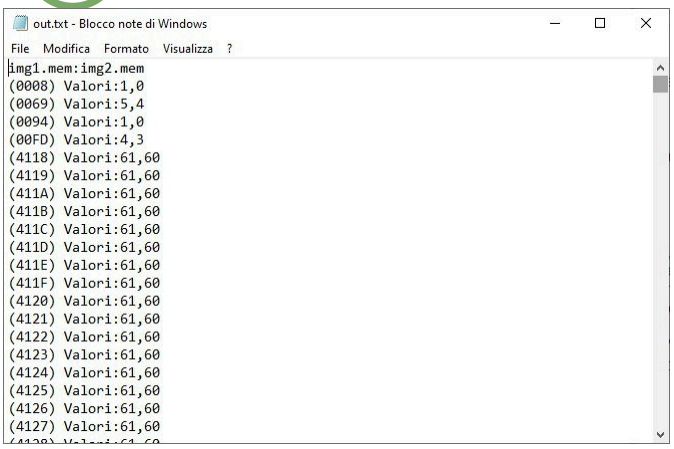


Figure 1

If we had a third snapshot, generated when the lives during the game had become 2, we could launch the tool again and check if the data continues decrease by 1.

Let's generate the third snapshot, we'll call it "img3.mem".
This time in the first text box, we'll put "img2.mem" and in the second box "img3.mem". As a result file I would opt for "out2.txt".
Let's check again (see figure 2).

As we can see, now the 00FD memory location contains the values we expected, namely 3 and 2. The outcome is perfect once again and we can therefore consider our starting point to be correct.

But our road is getting a little uphill now. We need to identify the area of the RAM where the content of this instruction is decreasing. Please also note that the 00FD address is located on page zero. The assembly instruction we are looking for could be a statement like: DEC $FD or C6 FD in machine code (hexadecimal notation). A simple search instruction inside the VICE monitor can provide us with further valuable clues. To do this, we then type:

h 0000 FFFF C6 FD (Enter)

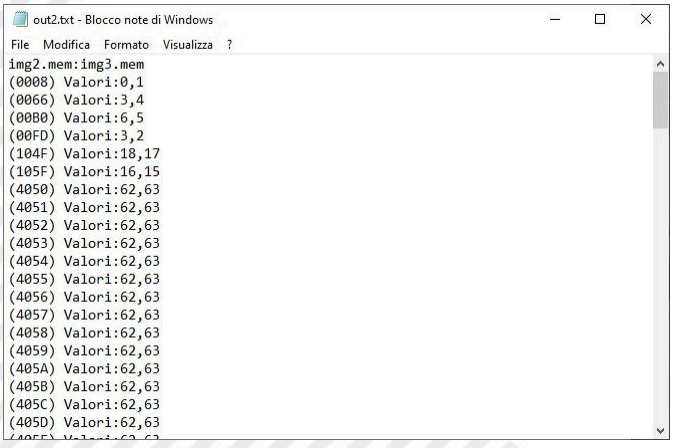


Figure 2

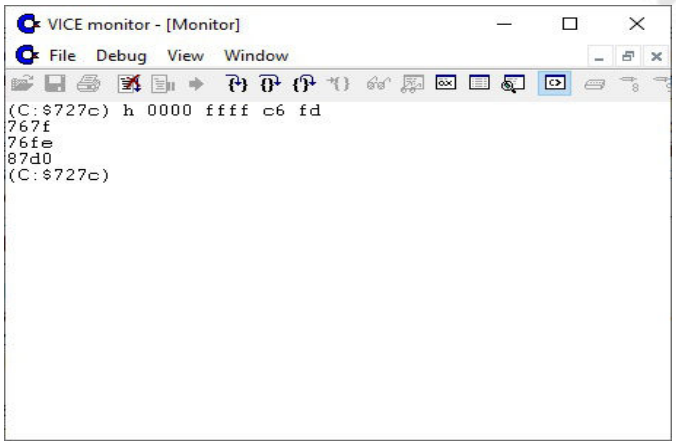


Figure 3

(see figure 3) and we get three addresses that could possibly contain the number of lives decreasing.

If we don't want to spend some time to carefully study portions of assembly code around these locations, we can simply try filling in 2 NOP (EA) opcodes for the two bytes at each of the three addresses that the search provided.

So, always using the VICE monitor, we type:

a 767f (Enter)
nop (Enter)
nop (Enter)
(Enter)

as shown in figure 4. Then we try to resume the game, exiting the VICE monitor. We get no tangible results, so we proceed in the same way, starting from the second instruction reported by the previous search.
So let's do this:

a 76FE (Enter)
nop (Enter)
nop (Enter)
(Enter)

and let's get back to play again.

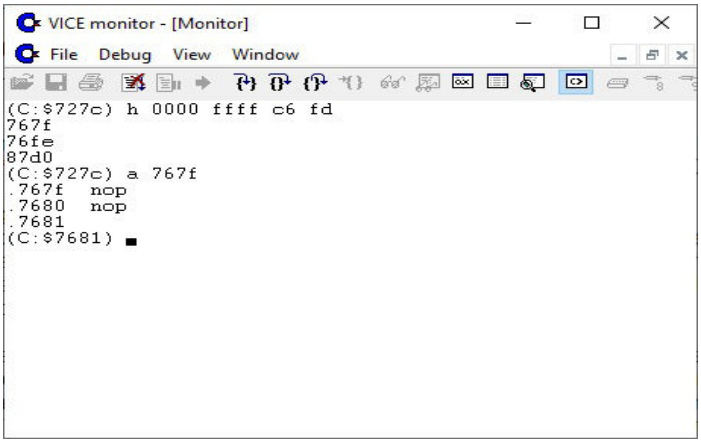


Figure 4

This time something rather strange (and certainly undesirable) occurs. As soon as we lose a life, the game ends!
Here is the logic that must guide us, along with a 6502/6510 assembly basic knowledge, it surely does not hurt.

What happens means that something changes in that memory area, just when we lose a life. Let's take a deeper look at the assembly code of that area.

So let's consider an area of memory that starts a few bytes before (76F4) and ends a few bytes after (7706) and let's disassemble that area by going back into the VICE monitor and typing:
d 76F4 7706 (Enter)

We get the result that you see in Figure 5.

Between the two NOP statements, that we added instead of the statement that would decrease the memory location with the number of lives, we can see two BPL statements (Branch if PLus or "Skip if positive"). Following an exquisitely logical line of thought, we could imagine that, during the course of the game, at the loss of a life, the control would be transferred to the DEC $FD instruction (which we removed by putting in place two NOP instructions that do absolutely nothing).

Then, if the number contained in this location had stayed positive, the jump would be made, as expected by the BPL $7717 instruction. It is likely that, by messing around that portion of the code, the information in the accumulator is no longer reliable. In fact, it is "dirty" now, as they say in technical jargon.
So what do we do? We unconditionally force a jump to that location. How?

Let's go into the VICE monitor and type:
a 76FE JMP $7717 (Enter)
(Enter)

as shown in figure 6. Now we can close the VICE

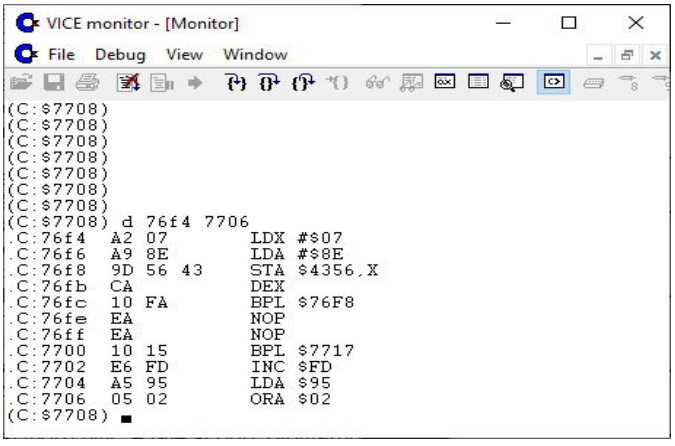


Figure 5

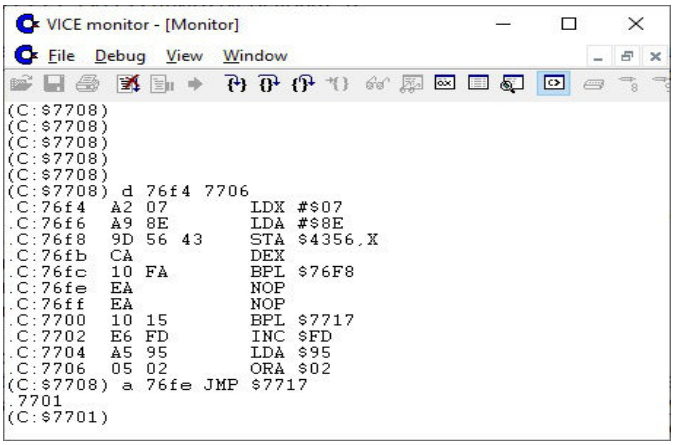


Figure 6

monitor and stat playing again. And magically we won't lose any more lives during the game! Summing up, in order to play infinite lives with this game, just open it with VICE, then open the VICE monitor and type:
a 76FE JMP $7717 (Enter)
(Enter)

Then we can get out of the VICE monitor to play regularly without losing any lives. The "crack" is served using a single assembly instruction, consisting of a total of 3 bytes!

Conclusions

This is one of those cases where, despite the technique allows you to locate the memory addresses containing the number of lives, it is necessary to know some assembly in order to achieve the desired result.

I hope that this article has turned your interest on and can be an incentive to experiment these ideas on your own with your favourite video games. Likewise, I'd love to get feedback on the fact that this article has made you reflect on the importance of knowing a bit of assembly language, so that this "magic" becomes more within your reach.

Greetings to all of you, dear readers and see you next time!

Binary File comparison tool (download):

http://www.retromagazine.net/download/TOOL_Confronta_immagini_memoria_vice.zip

RetroMath: Secret codes

by Giuseppe Fedele

The problem of transmitting secret messages is as old as writing. Encryption or cryptography (from the Greek kryptos, “hidden” and graphia, “writing") studies methods to encrypt a message so that it is not understood by people who are not authorized to read it.

The oldest example of cryptography dates back about 4500 years to the period of the Ancient Kingdom in Egypt and relates to some hieroglyphics carved in ancient monuments. Plutarch (46-48 AD – 125-127 AD), in Lysander's Life, describes the use of a “scytale” (Figure 1), a rudimentary cryptographic system that was exploited by Spartans, particularly in time of war, for short communications. The sscytale was a small wooden stick wrapped in a leather cord engraved with a message that could only be deciphered by having a stick the correct diameter of the one with which it had been written.

Fig.1 - Spartan scytale

It is well known that Julius Caesar (1st century BC) received information and transmitted orders to his lieutenants through a rudimentary coding system consisting of the translation (transliteration) of the letters of a message, known as the Caesar Code. In this code each letter is replaced by the following letter after n placed in the alphabet, imagined as a circular sequence, so after Z returns A. We fix a number from 0 to 25 (English alphabet) that will be the secret key K. The encryption operation using Caesar's Code consists of adding K to each character of the message in clear: the encryption, that is, is obtained by moving each character of the message in clear forward by K. If for example we put K=3 then the message:

m=’we will attack tomorrow’

becomes:

c=’zh zloo dwwdfn wrpruurz'

The Caesar Code is connected to modular arithmetic. Let a, b be two integers and n be a positive integer. So it is said that a and b are congruent module n if the difference a-b is a multiple of n. $a \equiv b \ (mod\ n)$

You can easily see that the following properties apply:

$$b \equiv a \ (mod\ n)$$
$$a - b \equiv 0 \ (mod\ n)$$

So, in the Caesar Code, the position of the encrypted letter corresponds to the formula:

$$c(x) = x + K \ (mod\ n)$$

where x is the position of the letter you want to encrypt, c(x) the position of the encrypted letter, K the key and n the number of characters of the alphabet. The message ‘DAZIO’, with key K=3, will then be encrypted in

- ‘D’, $x = 3$, $c(x) = 6 \ (mod\ 26) = 6$, corresponding to ‘G’
- ‘A’, $x = 0$, $c(x) = 3 \ (mod\ 26) = 3$, corresponding to ‘D’
- ‘Z’, $x = 25$, $c(x) = 28 \ (mod\ 26) = 2$, corresponding to ‘C’
- ‘I’, $x = 8$, $c(x) = 11 \ (mod\ 26) = 11$, corresponding to ‘L’
- ‘O’, $x = 14$, $c(x) = 17 \ (mod\ 26) = 17$, corresponding to ‘R’.

The message is decrypted using the reverse formula

$$c^{-1}(x) = x - K \ (mod\ n)$$

Caesar's possible digits in English are 25, so if a letter moves by 25 positions it returns to its starting point, thus matching the encrypted message with the clear message and making the operation useless. Given the small number of possibilities, the opponent who wants to force the code could try to decipher the message with all possible keys from K=1 to K=25 hoping to come across a full-fledged message.

Even the most elaborate digits based on correspondence between letters in the alphabet, even if varied periodically, are vulnerable because, when using the same key, identical letters are encoded with identical letters. You can then guess the letters most commonly used in a given language and trace the message back in plain language. Think of the Enigma machine (Figure 2) used by Germans during World War II. Taking advantage of the statistical frequency of German alphabetical letters, the group of English scientists (mathematicians, linguists and cryptographers) led by Alan Turing (Figure 3) managed to force the encryption algorithm.

Fig.2 - Enigma encryption machine

The Enigma machines in the army version initially had three rotors that could be extracted and replaced. The first task for an Enigma operator was to decide where each individual rotor should be set. There were five rotors to choose from, they could be inserted into the three Enigma housings. The configuration of the German machines was changed every

24 hours following a certain protocol to be kept in total secrecy because, if the allies had possession of them, they could easily decipher each message. The only way to decipher the encryption was therefore to have an Enigma machine configured exactly as the one with which the message had been encrypted.

Unlike the described ciphers in which one character is replaced by another, in 1929 mathematician Lester S. Hill designed an encryption algorithm based on modular

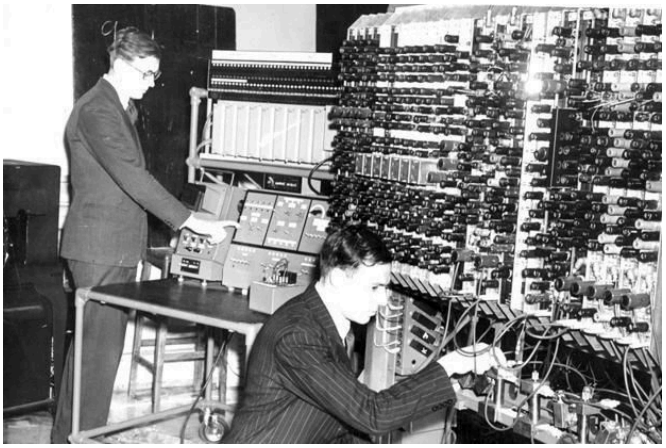


Fig.3 - Alan Turing

arithmetic and linear algebra. The idea is to divide the plain text into groups of two letters by associating the corresponding position from 0 to 25 with each letter. The "CIFRARIO DI HILL" message will then be composed by the table

CI	FR	AR	IO	DI	HI	LL
2,8	5,17	0,17	8,14	3,4	7,8	11,11

To encrypt the message, we use a matrix

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with a determinant equal to 1, i.e. ad-bc=1. In this case the inverse matrix is

$$M^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

So if we choose how $M = \begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}$

its inverse is: $M^{-1} = \begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix}$

The chosen matrix and its inverse represent the encoding and decoding key respectively.

For each pair of letters/digits the coding matrix is multiplied by the associated pair of numbers and the result is translated in the range 0-25 through the module operations; for the "CI" pair

$$\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 2 \\ 8 \end{pmatrix} = \begin{pmatrix} 46 \\ 18 \end{pmatrix} \ (mod\ 26) = \begin{pmatrix} 20 \\ 18 \end{pmatrix}$$

The "CI" pair is then encoded in "US". Similarly, the "FR" pair is encoded in "WN" because

$$\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 5 \\ 17 \end{pmatrix} = \begin{pmatrix} 100 \\ 39 \end{pmatrix} \ (mod\ 26) = \begin{pmatrix} 22 \\ 13 \end{pmatrix}$$

To decipher the message, proceed in the same way considering the decoding matrix. The encrypted pair "US" will be decrypted as

$$\begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} 20 \\ 18 \end{pmatrix} = \begin{pmatrix} -50 \\ 34 \end{pmatrix} \ (mod\ 26) = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$$

while the "WN" pair

$$\begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} 22 \\ 13 \end{pmatrix} = \begin{pmatrix} -21 \\ 17 \end{pmatrix} \ (mod\ 26) = \begin{pmatrix} 5 \\ 17 \end{pmatrix}$$

The problem with all these encryption/decryption methods is that if A wants to send a message to B, it must first tell B which secret key to use. So the key should also be transmitted so that it is not intercepted by prowlers.

To overcome this drawback, in 1978 three professors: Ronald Rivest, Adi Shamir and Leonard Adleman of MIT carried out a mathematical calculation procedure that will be called the "RSA algorithm", from the initials of its inventors. The RSA algorithm is based on some properties of prime numbers (i.e. positive integers greater than 1 divisible only by 1 and by itself; example prime numbers are: 2,3,5,7,11,13,17,19,23,29,...). One property is that, given two prime numbers, it is very easy to determine their product, whereas it is much more difficult to determine, from a given number, which prime numbers produced that result after being multiplied with each other.

How does RSA work? Let's give a simple example

Giuseppe must send Marco a text (coded for example by placing each letter in the alphabet: 0-25).

1. Marco generates two prime numbers p and q and multiplies them with each other obtaining the number N that is made public whereas the two numbers p and q must remain secret (for p and q very large numbers are used, in the example we consider small numbers):

$p = 5, q = 11$	N=55

2. Marco finds the value of Euler's function and stores the value in b. Euler's function is defined for each positive integer n and is equal to the number of integers between 1 and n that are co-prime with n $\Phi(N)$ (two numbers x and y are co-prime if they do not have dividers in common, i.e. if their maximum common divider is 1). This function proves to be

$$\Phi(N) = (p-1)(q-1)$$

$\Phi(55) = (5-1)(11-1) = 40$	b=40

3. Marco calculates the first integer c that is co-prime with b

$c = 2$	$MCD(2,40) = 2$	NO	c=3
$c = 3$	$MCD(3,40) = 1$	SI	

c is the second public key.

4. Marco calculates the number d such that

$$d \times c \equiv 1 \pmod{b}$$

$d = 2$	$2 \times 3 \ (mod\ 40) =$	6	NO	d=27
$d = 3$	$3 \times 3 \ (mod\ 40) =$	9	NO	
$d = 4$	$4 \times 3 \ (mod\ 40) =$	12	NO	
⋮	⋮	⋮	⋮	
$d = 26$	$26 \times 3 \ (mod\ 40) =$	38	NO	
$d = 27$	$27 \times 3 \ (mod\ 40) =$	1	SI	

d is the key to decipher and must stay secret.

To transmit a message to Marco, Giuseppe prepares the string m consisting of q letters and associates with each letter a numeric value (for example, the position of the letter in the English alphabet, 0-25):

$$m = (m_1, m_2, \dots, m_q)$$

Then Giuseppe reads Marco's public keys N and c and transmits the numbers of m one at a time encrypting them as

$$h_k = m_k^c \ mod\ N, \quad k = 1, \dots, q$$

For example, to transmit the number 12, Giuseppe calculates and transmits

$$h = 12^3 \ mod\ 55 = 23$$

When Marco receives the digit h=23, he uses the secret key d to decrypt the message

$$h^d \ mod\ N = 23^{27} mod\ 55 = 12$$

The listing in the box implements the steps described in the example on a lowercase character string and uses the Pascal online compiler, which can be found at:

https://www.onlinegdb.com/online_pascal_compiler

The clearly simplified code does not have a whole series of controls to handle the various exceptions. It implements various functions including pot(x,y) that performs power elevation between x and y integers. Consider that the maximum allowable integer is MAXINT=32767, so could overflow (I recommend choosing values of prime m_k^c numbers suitable for our retro-computers!).

The MCD function (x,y) takes advantage of the property that if two numbers x and y are divisible by a third number z, then their difference will also be:

$$x = kz, \quad y = nz \quad \rightarrow \quad x - y = (k - n)z$$

and this explains why calculating MCD between x and y is the same as calculating MCD between x-y and y or between x and y-x:

$$MCD(x, y) = MCD(x - y, y) = MCD(x, y - x)$$

Repeated subtractions necessarily lead to a time when x=y, that is, there is no difference. At this point it means that both the first and second numbers are divisible by x, which is therefore the MCD sought.

The other functions implement the various steps envisaged in the encryption phase by the algorithm.

Bibliography:
[1] Alko R. Meijer. Algebra for Cryptologists. Springer, 2016.
[2] S. Rubinstein-Salzedo. Cryptography. Springer, 2018.
[3] K. Martin. Everyday Cryptography. Oxford University Press, 2017.

```
program RSA;
var p,q : INTEGER;
var N,b,c,d : INTEGER;
var i : INTEGER;
var str : STRING[10];
var mk,he : INTEGER;
function pot(x,y:INTEGER):INTEGER;
var p,i:INTEGER;
begin
  p:=1;
  for i:=1 to y do
    p:=p*x;
  if y>=0 then
    pot:=p
  else begin
    pot:=0;
  writeln('Error!!!');
  end
end;
function MCD(a,b:INTEGER):INTEGER;
begin
  while a<>b do
    if a>b then
      a:=a-b
    else
      b:=b-a;
  MCD:=a;
end;
function EulerPhi(p,q:INTEGER):INTEGER;
begin
EulerPhi:=(p-1)*(q-1);
end;
function firstPublicKey (p,q:INTEGER):INTEGER;
begin
firstPublicKey:=p*q;
end;
function secondPublicKey(b:INTEGER):INTEGER;
var c:INTEGER; find:BOOLEAN;
begin
  find:=false;
  c:=2;
  while not find do
    if MCD(c,b)=1 then
      find:=true
    else
      c:=c+1;
  secondPublicKey:=c;
end;
function decryptKey(b,c:INTEGER):INTEGER;
var d:INTEGER; find:BOOLEAN;
begin
  find:=false;
  d:=2;
  while not find do
    if (d*c) mod b=1 then
      find:=true
    else
      d:=d+1;
  decryptKey:=d;
end;
function encryptChar(mk,c,N:INTEGER):INTEGER;
begin
  encryptChar:=pot(mk,c) mod N;
end;
begin
  p := 5;
  q := 11;
  N := firstPublicKey(p,q);
  b := EulerPhi(p,q);
  c := secondPublicKey(b);
  d := decryptKey(b,c);
  writeln('----------------------');
  writeln('N = ',N);
  writeln('Phi(N) = ',b);
  writeln('c = ',c);
  writeln('d = ',d);
  writeln('----------------------');
  str:='giuseppe';
  writeln(str);
  for i:=1 to length(str) do begin
    mk:=ord(str[i])-ord('a');
    he:=encryptChar(mk,c,N);
    writeln(str[i],' ',mk,'->',he);
  end;
end.
```

3D Graphs with a few lines in BASIC

by Alberto Apostolo

Everyone in Italy knows Alberto Angela as a skilled conductor of television broadcasts of science dissemination and general culture. But those who lived in the 70s and 80s of the 20th century also remember his illustrious parent: Piero Angela, the popularizer par excellence.

Piero Angela conducted from 1981 to 1994, "Quark", a program broadcast by R.A.I. The theme song for "Quark" was taken from the Bach's Air On the G String with the animated graph (Figure 2).

Figure 2 - https://www.youtube.com/watch?v=8CW0GfTdMHM

Although it will not be exactly like that of "Quark", with some care we can also build a graph of a function z=f(x,y) and perhaps add some useful tools in the study of functions in two variables that meet in Advanced Calculus.

From theory to practice

[MS91] describes in Appendix a program for obtaining the graph of z=f(x,y) defined on the subset [-1,1]X[-1,1] of the XY plane.

This program was used as a starting point by the undersigned to add subsequent changes and use the subset [-A,A]X[-A,A] where A is a real number greater than zero.

The graph is represented by appropriately plotting curves having an equation

$$\begin{cases} x=x_i=\text{cost.}, -1\leqslant y\leqslant +1, i=0,\ldots,40 \\ z=f(x,y)\ . \end{cases}$$

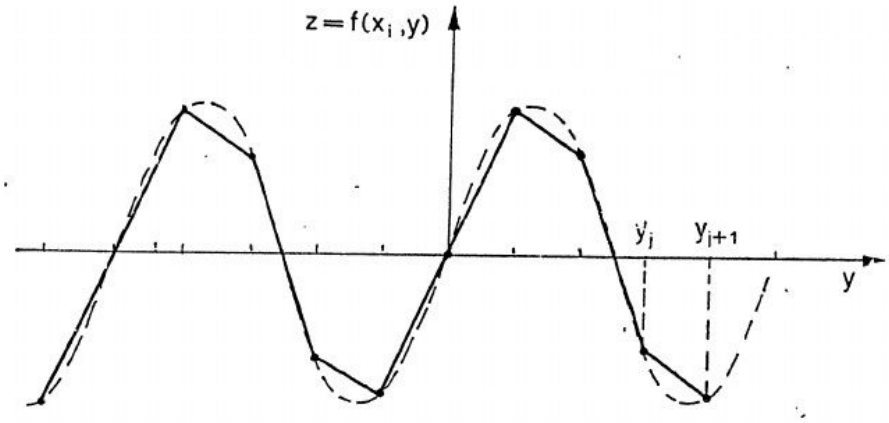


Figure 3

41 curves are plotted on planes parallel to the plane yz (the value 40 was set by the authors of the book).
The closest curve to the reader has x_0 = A while the farthest curve has x_40=-A (for reasons related to the fact that the closest curves hide the farthest ones).

Every single curve

$$z=f_i(y)=f(x_i,y).$$

is represented by dividing the subset [-A,A] into 160 equidistant ranges, with linear interpolation in each range as in the example in Figure 3 [MS91] (also 160 is a value set by the authors of the book).

The three-dimensional effect is given by the overlap of two successive graphs by translating the reference (O,y,z) as in the example in Figure 4 (always taken from [MS91]), without tracing the parts of curves hidden from those in the foreground.

This is why you enter a coordinate system (u,v) to consider the computer screen and try to know the relationship to switch from coordinates (x,y,z) to coordinates (u,v).

On screen the square

[-A,A]X[-A,A]
(viewed with three-dimensional effect) would have the shape of a parallelogram (Figure 5) with the coordinates calculated according to the rules of the GWBASIC in 640x200 points graphical mode (where the origin is at the top left).
The relationship between the pair (u,v) and the coordinates (x,y,z) expressed in matrix form is as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

e.g.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y + a_{13}z + b_1 \\ a_{21}x + a_{22}y + a_{23}z + b_2 \end{pmatrix}.$$

To determine b1 and b2 it is necessary that the point (0.0.0) has screen coordinates (290.100), a choice dictated by personal favorites and the need to "center" the graph a little.

The coefficients a13 and a23 are obtained by imposing that the point (0,0,1) is at the coordinates (290,100-K) where K is a coefficient of expansion of the z-axis in order not to have images too crushed.

The remaining values a11, a12, a21, a22 are obtained by "clever" conditions, i.e. using the points (A,0,0) and (0,A,0).

In the end you will have the formulas:

```
U1 = 290 + (-80 * X + 160 * Y) / A
V1 = 100 + (40 * X) / A
V = V1 - Z * K
U = U1
```

Now suppose we have already plotted n curves and want to plot the n+1 curve.
The max(u) function is defined, representing the maximum of the v ordinates plotted by the first n curves and the min(u) function representing the minimum (Figure 6).
When plotting the n+1 curve, the "visible" parts must have v sorted out of range (min(u), max(u)) for each u belonging to the domain of the n +1

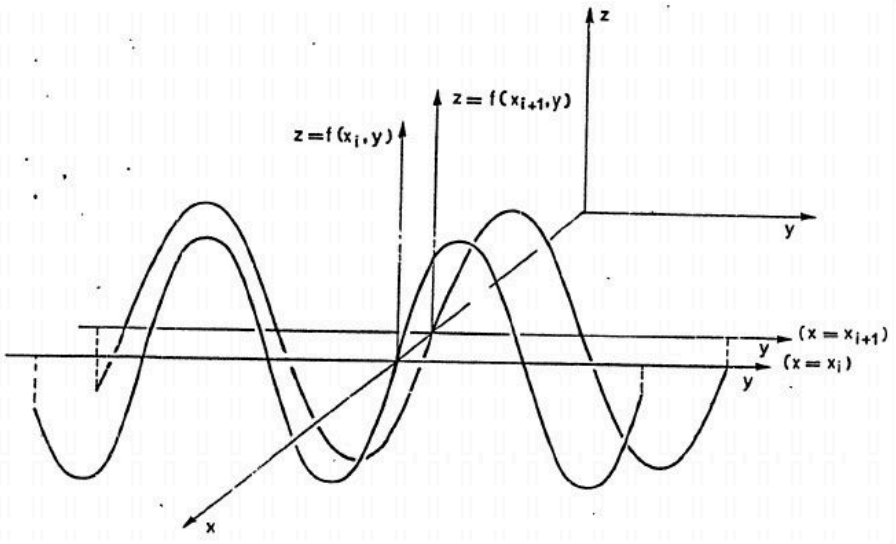


Figure 4

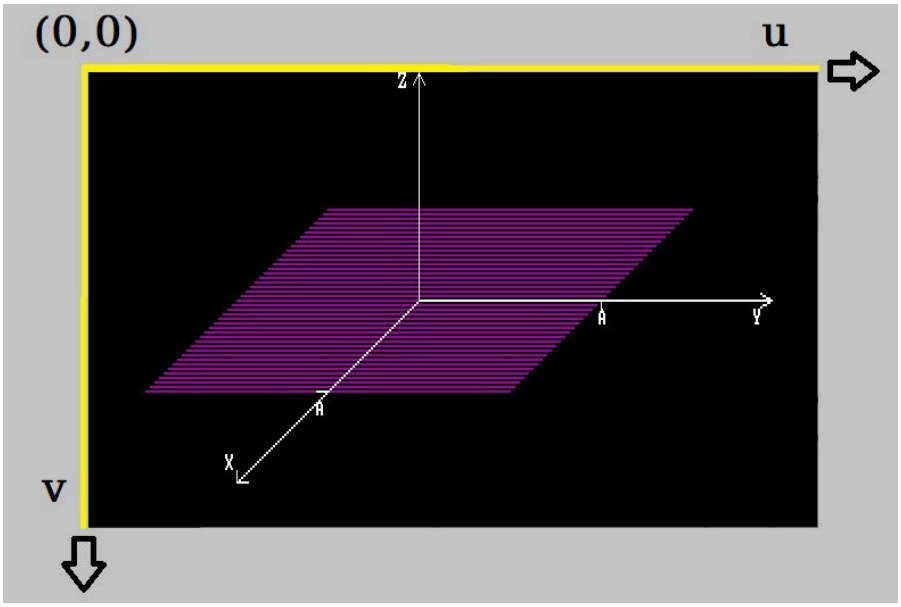


Figure 5

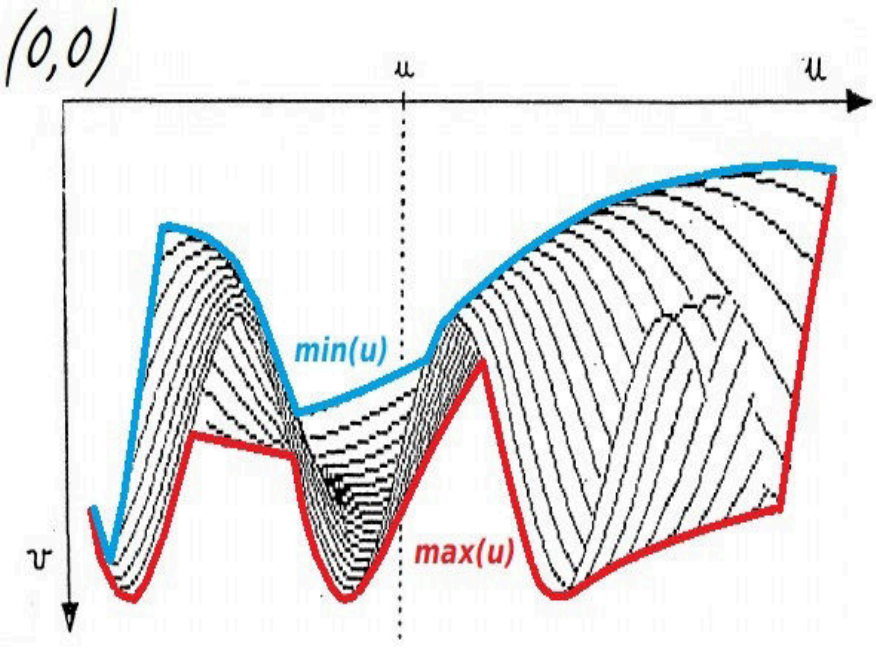


Figure 6

curve.
The program GRAF3D01.BAS in GWBASIC condenses all the considerations made so far (Figure 7). Program lines are briefly described below.

Lines 1000-1030:
variables declaration.

Lines 1040-1080:
screen preparation and colour palettes used.

Lines 1090-1120:
initializing parameters. Variable A refers to the square width of the xy plane on which to represent the function, K is a scale coefficient for the z-axis, CONSTRAINTS$ ="Y" indicates that there are conditions placed on the values of x and y (the values of x and y not included are highlighted in yellow), CHKONLYSSET$ ="Y" indicates that you only want to study the existence of the function (in this case you set z=0, where the function exists the color is purple otherwise it is green).

Lines 1130-1180:
start of processing, declaration of iterative cycles (for abscesses and ordinates), program line declaration to skip to in case of error in the calculation of the function, use of purple color for the graph.

Lines 1190-1220:
program area where the conditions on the x and y points are applied. Instructions can be inserted / deleted / modified between lines 1201 and 1219.

Lines 1230-1250:
program zone where to calculate the function z=f(x,y). Instructions can be inserted / deleted / modified between lines 1231 and 1259.

Lines 1260-1270:
handling function existence study.

Lines 1280-1290:
handling function calculation error. For any error, do not stop the program but follow the instructions below.

```
1000 REM GRAF3D01.BAS
1010 DEFINT C, I-J, U-V
1020 PI = 3.1415926654#
1030 DIM MAX(640), MIN(640)
1040 REM SCREEN 16 COLORS 640 * 200 PIXELS
1050 REM 0=BLACK,5=PURPLE,14=YELLOW,15=WHITE,4=GREEN
1060 SCREEN 8: DATA 0,5,14,15,3: FOR I = 0 TO 4: READ C: PALETTE I, C: NEXT I
1070 REM CLEAR SCREEN
1080 KEY OFF: CLS
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "N": CHKONLYSSET$ = "N"
1110 A = 6.29 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
1120 K = 50: REM FACTOR FOR Z
1130 REM PROCESSING
1140 X = A: Y = -A
1150 FOR I = 0 TO 40
1160 FOR J = 0 TO 160
1170 ON ERROR GOTO 1280
1180 KOLOR = 1
1190 REM CONSTRAINTS ON F(X,Y)
1200 IF CONSTRAINTS$ <> "Y" THEN GOTO 1230
1210 IF ((X ^ 2 + Y ^ 2) <= 2) THEN GOTO 1230
1220 KOLOR = 2: Z = 0: GOTO 1290
1230 REM TYPE THE FUNCTION Z=F(X,Y) HERE
1240 T = X * X + Y * Y
1250 Z = EXP(-T/8)*COS(T)
1260 IF CHKONLYSSET$ = "Y" THEN Z = 0
1270 GOTO 1290
1280 IF ERR > 0 THEN Z = 0: KOLOR = 4: RESUME 1290
1290 ON ERROR GOTO 0
1300 REM SCREEN COORDINATES
1310 U1 = 290 + (-80 * X + 160 * Y) / A: V1 = 100 + (40 * X) / A
1320 IF ((V <> V1) OR (Z <> 0)) AND (J > 0) THEN KOLOR = 1
1330 V = V1 - Z * K: U = U1 : REM PROJECTION OF COORDINATE Z
1340 IF I = 0 OR J = 159 OR J = 160 THEN MAX(U) = V: MIN(U) = V: GOTO 1380
1350 IF V > MAX(U) THEN MAX(U) = V: GOTO 1380
1360 IF V < MIN(U) THEN MIN(U) = V: GOTO 1380
1370 C = 0: GOTO 1400
1380 IF C = 0 OR J = 0 THEN PSET (U, V), KOLOR
1390 LINE -(U, V), KOLOR: C = 1
1400 Y = Y + 2 * A / 160: NEXT J
1410 X = X - 2 * A / 40: Y = -A: NEXT I
1420 DATA 22,16,"X",14,74,"Y",1,35,"Z"
1430 FOR I = 1 TO 3: READ X, Y, A$: LOCATE X, Y: PRINT A$: NEXT I
1440 LOCATE 14, 57: PRINT A: LOCATE 19, 26: PRINT A
1450 DATA 290,100,0,-100,290,0,-5,5,290,0,5,5
1460 DATA 290,100,310,0,600,100,-10,-3,600,100,-10,3
1470 DATA 290,100,-160,80,130,180,0,-5,130,180,10,0
1480 DATA 210,140,-10,0,450,100,0,3
1490 FOR I = 1 TO 11: READ X, Y, U, V: PSET (X, Y), 3: LINE -STEP(U, V), 3
1500 NEXT I
1510 IF INKEY$ = "" THEN GOTO 1510
1520 END
```

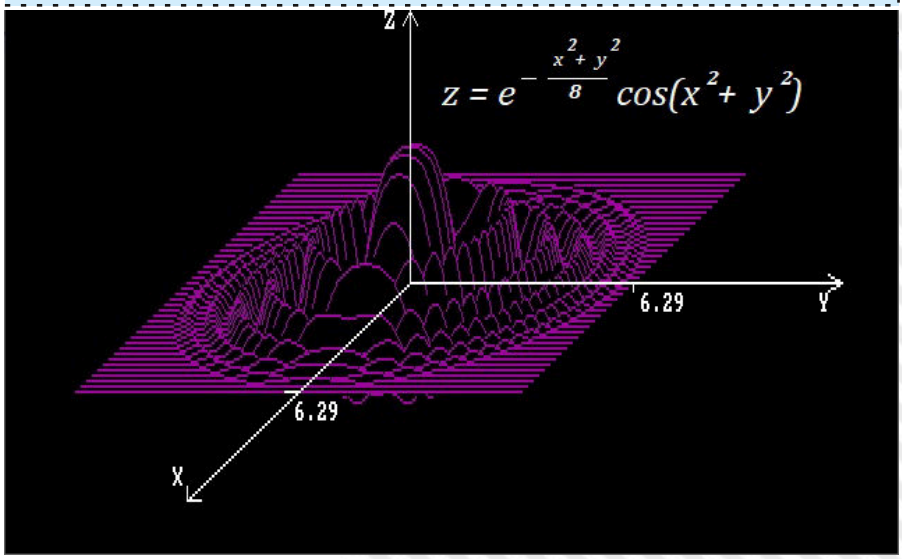


Figure 7

Lines 1300-1330:
calculation of screen coordinates.

Lines 1340-1370:
checks whether or not a part of the curve is to be represented.

Lines 1380-1390:
draw a point and the straight line connecting it to the previous point.

Lines 1400-1410:
increment x and y values and close the respective iterative blocks.

Lines 1420-1500:
drawing of the Cartesian axes (in white) with abscissa and ordered A.

Lines 1510-1520:
endless wait for a button pressed and end of program.

Figure 8 shows the study of the existence of a function in a certain region of the xy plane. By modifying some lines of the program you can study z=sqr(x*y) at the points inside the circle of radius 1 (including the border).
The paragraph ends by showing how to "cut" the graph. Figure 9 shows the graph of the raised function of 1 (to also observe the negative part). Condition x <= 0 is imposed on line 1210 to have the section with the plane x = 0.
The best sections are obtained for x=0 or with planes parallel to the plane x=0. In other cases, side effects are obtained by using curves f(x_i,y) parallel to the plane yz (Figure 10) to plot the graph.

Rotating charts
The program GRAF3F01.BAS always offers the same perspective. You can add the ability to "rotate" the chart about the Z axis to have another point of view and observe details previously escaped . To achieve what is desired, the axes rotation formulas must be applied to the coordinates (x,y) for a certain theta angle such that positive angles correspond to counterclockwise rotations:

$$xt = +x\cos\theta + y\,\mathrm{sen}\,\theta\ ,$$
$$yt = -x\,\mathrm{sen}\theta + y\cos\theta\ .$$

```
1000 REM GRA3D01.BAS
...
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "Y": CHKONLYSSET$ = "Y"
1110 A = 1 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
...
1190 REM CONSTRAINTS ON F(X,Y)
...
1210 IF ((X ^ 2 + Y ^ 2) <= A) THEN GOTO 1230
...
1230 REM TYPE THE FUNCTION Z=F(X,Y) HERE
1240 Z = SQR(X*Y)
...
1520 END
```

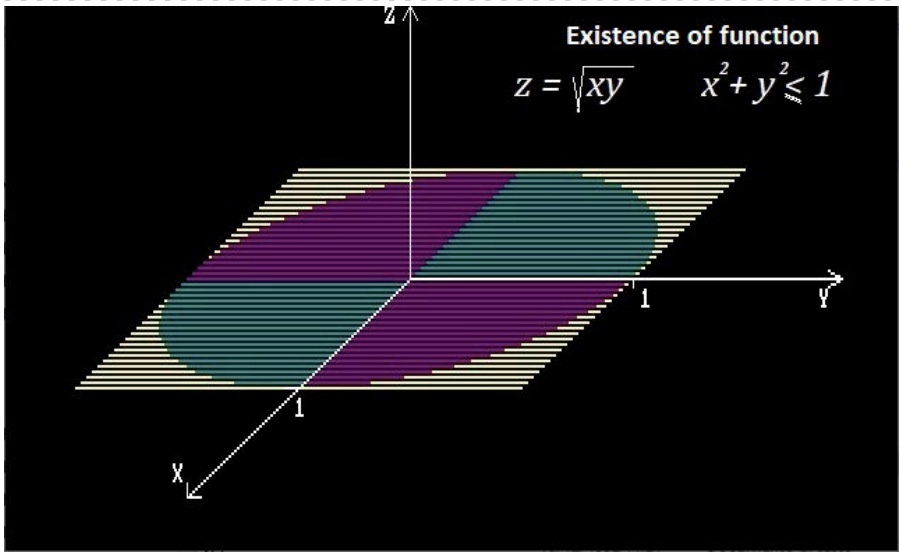


Figure 8

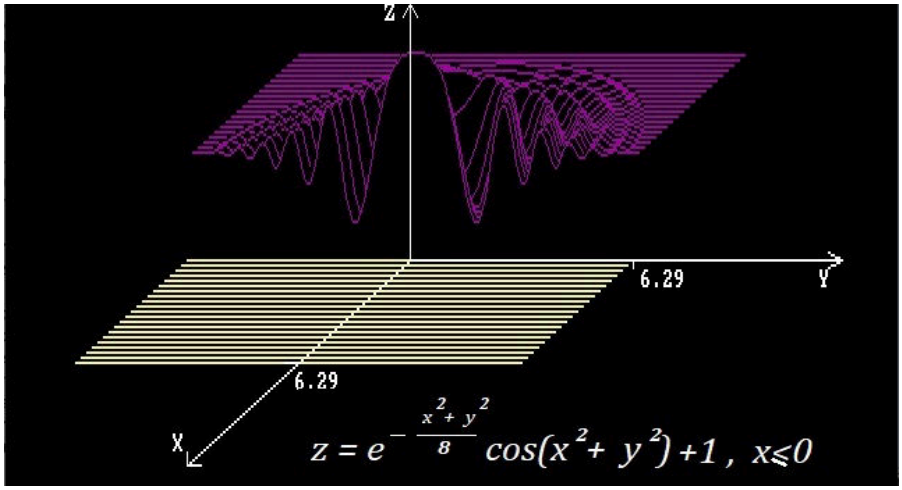


Figure 9

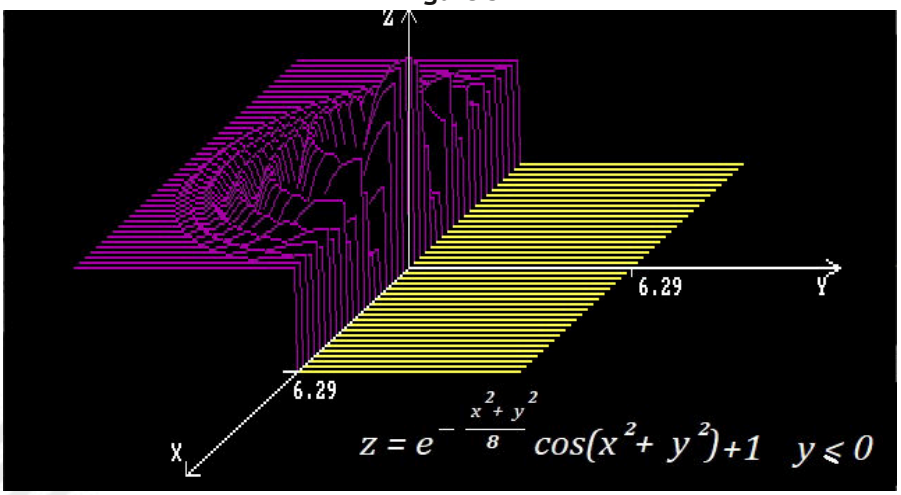


Figure 10

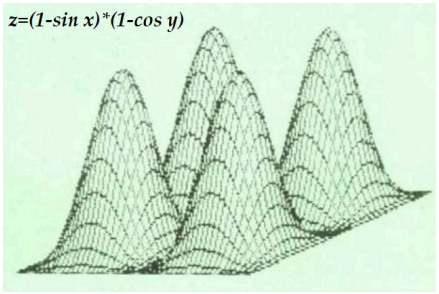
Then the triplets (x,y,f(xt,yt)) are returned to the screen to give the rotational effect of the graph. The new coordinates (xt,yt) must also be applied to the constraints.

Program GRAF3D02.BAS is what we get (Figure 11, where bold-italics highlight the differences compared to GRAF3D01.BAS).

Conclusions

The method dealt with in this article is not the only one.

Another approach was used by Belloni and Giacinti [BG84] for their program on ZX Spectrum, which takes advantage of the versatility of the graphical instructions of the ZX Spectrum and implements an algorithm capable of erasing surface lines f(*,y) and f(x,*) already drawn (to replace them with those of the foreground lines) without considering cavities and intersections (as in the figure below).



Bibliography

[BG84] A. Belloni, E. Giacinti, "TREDIM: grafica a tre dimensioni", MC Microcomputer n.29, Aprile 1984, pag.118-120. https://issuu.com/adpware/docs/mc029

[MS91] P. Marcellini, C. Sbordone, "Esercitazioni of Matematica", Vol. 2, Parte 2, Liguori Editore, 1991.

```
1000 REM GRA3D02.BAS
1010 DEFINT C, I-J, U-V
1020 PI = 3.1415926654#
1030 DIM MAX(640), MIN(640)
1040 REM SCREEN 16 COLORS 640 * 200 PIXELS
1050 REM 0=BLACK,5=PURPLE,14=YELLOW,15=WHITE,4=GREEN
1060 SCREEN 8: DATA 0,5,14,15,3: FOR I = 0 TO 4: READ C: PALETTE I, C: NEXT I
1070 REM CLEAR SCREEN
1080 KEY OFF: CLS
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "Y": CHKONLYSSET$ = "N"
1110 A = 4 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
1120 K = 40: REM COEFFICIENT FOR Z
1125 THETA=-PI/4 : ROTATION OF AXIS X,Y
1126 DEF FNXT(X,Y,THETA)=X * COS(THETA) + Y * SIN(THETA)
1127 DEF FNYT(X,Y,THETA)=-X * SIN(THETA) + Y * COS(THETA)
1130 REM PROCESSING
1140 X = A: Y = -A
1150 FOR I = 0 TO 40
1160 FOR J = 0 TO 160
1170 ON ERROR GOTO 1280
1180 KOLOR = 1
1185 XT = FNXT(X,Y,THETA):YT = FNYT(X,Y,THETA)
1190 REM CONSTRAINTS ON  F(XT,YT)
1200 IF CONSTRAINTS$ <> "Y" THEN GOTO 1230
1210 IF (YT<=0) THEN GOTO 1230
1220 KOLOR = 2: Z = 0: GOTO 1290
1230 REM TYPE THE FUNCTION Z=F(XT,YT) HERE
1250 Z = SQR(1-XT*XT/4 -YT*YT/9)
1260 IF CHKONLYSSET$ = "Y" THEN Z = 0
1270 GOTO 1290
1280 IF ERR > 0 THEN Z = 0: KOLOR = 4: RESUME 1290
1290 ON ERROR GOTO 0
1300 REM SCREEN COORDINATES
1310 U1 = 290 + (-80 * X + 160 * Y) / A: V1 = 100 + (40 * X) / A
1320 IF ((V <> V1) OR (Z <> 0)) AND (J > 0) THEN KOLOR = 1
1330 V = V1 - Z * K: U = U1 : REM PROJECTION OF COORDINATE Z
1340 IF I = 0 OR J = 159 OR J = 160 THEN MAX(U) = V: MIN(U) = V: GOTO 1380
1350 IF V > MAX(U) THEN MAX(U) = V: GOTO 1380
1360 IF V < MIN(U) THEN MIN(U) = V: GOTO 1380
1370 C = 0: GOTO 1400
1380 IF C = 0 OR J = 0 THEN PSET (U, V), KOLOR
1390 LINE -(U, V), KOLOR: C = 1
1400 Y = Y + 2 * A / 160: NEXT J
1410 X = X - 2 * A / 40: Y = -A: NEXT I
1420 DATA 22,16,"X",14,74,"Y",1,35,"Z"
1430 FOR I = 1 TO 3: READ X, Y, A$: LOCATE X, Y: PRINT A$: NEXT I
1440 LOCATE 14, 57: PRINT A: LOCATE 19, 26: PRINT A
1450 DATA 290,100,0,-100,290,0,-5,5,290,0,5,5
1460 DATA 290,100,310,0,600,100,-10,-3,600,100,-10,3
1470 DATA 290,100,-160,80,130,180,0,-5,130,180,10,0
1480 DATA 210,140,-10,0,450,100,0,3
1490 FOR I = 1 TO 11: READ X, Y, U, V: PSET (X, Y), 3: LINE -STEP(U, V), 3
1500 NEXT I
1510 IF INKEY$ = "" THEN GOTO 1510
1520 END
```

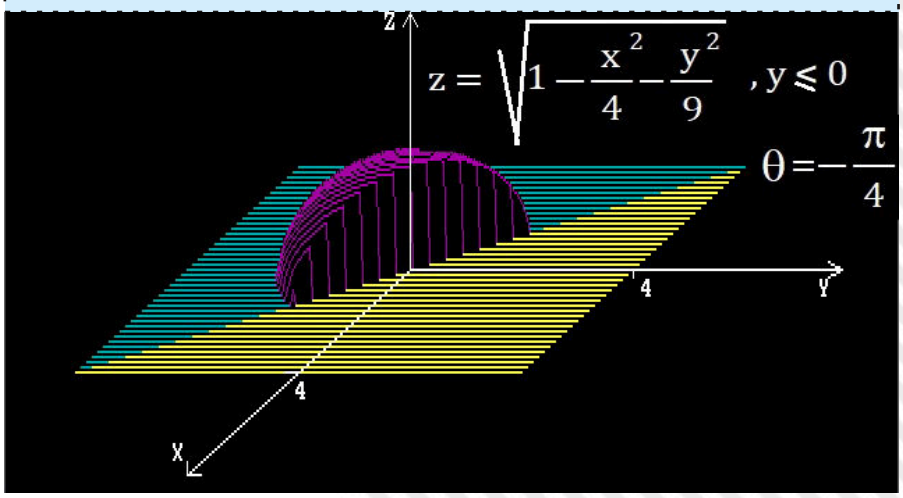


Figure - 11

Japan cronicles:
A new Game & Watch?

by Michele Ugolini

Welcome back dear readers, here's the review of Nintendo's new Game & Watch. The title is correct, this beloved electronic marvel has infected so many customers that it could not go unnoticed during the last decade. The last ten years have been really prolific in reproducing mini remakes, ornaments not always coherent and more or less successful dust-catching gizmos.

Rising from its ashes, awakened in 2009 from a 30 years nap, meet Ball, one of the world's most beloved entries in the Nintendo Game & Watch series. (see Figure 1)

2009 was a really interesting year at Nintendo. The president was Mr Iwata (Sapporo, December 6, 1959 – Kyoto, July 11, 2015), who was a very productive video game author and Japanese entrepreneur.

He was Nintendo's fourth president, serving from 2002 to 2015, after the longstanding predecessor Hiroshi Yamauchi (an acquaintance of ours, we me him in the tenth episode of this series).

We can read an interesting interview to Mr Iwata at this link:
https://www.nintendo.it/Iwata-Chiede/Iwata-Chiede-Game-Watch/Iwata-Chiede-Game-Watch/5-Ricreare-Ball-30-years-after/5-Ricreare-Ball-30-years-after-2233202.html

It was great fun to read what Iwata and his staff said during the interview. How many weird tricks have been put together during adventurous circumstances to be able to produce this collector's item again.

Let's take president Iwata and the entire staff: Izushi, Kano, Yamamoto and make them discuss the problem. A rare breed of people, so brilliant that they can shake the Earth: state-of-the-art genius and inventiveness! (see Figure 2)

For example, during the interview they talk about the new 2009 Ball, coming 30 years after the original, made as a limited edition gift for 2009 Platinum members of the Nintendo Club.

Nintendo Club is Nintendo's loyalty program which started in 2003 in Japan. In the UK, the Nintendo Club was officially opened in December 2007. Members were awarded Star points when registering purchased Nintendo consoles and software. Stars can be exchanged for the original Nintendo products available in the Nintendo Club Star Catalogue.

Why was Ball chosen over any other Game & Watch? In the interview it, Ball is called "the favorite game of our beloved Yokoi" (Gunpei Yokoi, see episode 10 of this series).

What were the major problems they had to face to recreate Ball?
The strategy to achieve the goal was clear in everybody's mind, but unlike the bizarre situation Yokoi had to face (described in the previous episode), this time it was necessary to adapt the constructive logic by taking from another dimension

Figura 1

all the knowledge and adapting it to a present far from the dawn of electronics in the 1980s.

So there were two main problems: retrieving both hardware and software. Being a game created 30 years before, there was no longer any documentation for it. It was difficult to contact all the people involved and collect the material.

It was also difficult to retrieve all the individual Game & Watch pieces, as they haven't existed in so many years. Then you had to retrieve all the diagrams and the shapes. They had to recreate everything from scratch.

Mr. Kano, working on the software, had to retrieve all the know-how of many years ago and in terms of hardware he had to recreate the feeling of pressing the keys and the gameplay... all based on distant memories!

When Mr Kano's collaborators asked him, "Why is this so?", he himself, with great humility, could only answer, "I have no idea!" After thirty years his memories had naturally faded.

Mr. Yamamoto, also part of Ball's rebirth group, ironically confessed this about Mr. Kano: "Good thing he did it when he still remembered it. Any minute he could forget everything, even while he was doing it!"
A really funny anecdote: do you know the two black pillars on the sides of the Mr Game & Watch character? Those two stripes are actually hiding something. In Ball's original game, on the edge of the screen, there were unnecessary electrical circuits that were covered to make them invisible. It was decided to hide them with a printed filter. In today's LCDs, you don't see the electrical circuits, so there's no need to hide them. The original was however recreated as faithfully as possible by including the pillars anyway (see Figure 3).

Let's mention one of the improvements. In the last episode of this series I explained how extremely important silence is on trains in Japan, and above all why the businessmen of the 1980s could not play a video game. Well, the only improvement made in Ball 2009 was the ability to turn off the audio. Nothing more!

In this review I want to give maximum praise to the adorable remake dated 2009. I will describe exactly how it works, also because, I admit, I occasionally dust off my Ball remake for a quick game. I'm not going to talk about the price. It is a coveted object, although not very rare and unfortunately to date, in 2020, the price no longer reflects any coherent logic.

In Ball, a juggler throws balls into the air and must retrieve them without dropping them, in an increasingly faster progression of gameplay. That's it! Here's the fun!

There are only two buttons used during the game. The button on the left side of the screen moves both of the character's arms to the left, while the button on the right side of the screen moves both of the character's arms to the right. The goal of the game is to juggle between the balls and make sure they don't hit the ground. This can be done by placing the character's hands under the balls. Once the hand touches a ball, that will be thrown upwards and curve to the other side of the character's body where the user must move the hand to catch it.

Figura 2

When you drop a single ball, the game ends and the player has to start again. In game A, the player juggles between two balls and receives one point for each successful catch, while in game B they are required to manage three balls and receive ten points for each catch. The highest possible score is 9,999 for game A and 9,990 for game B, since 9,999 is not a multiple of 10.

The player can easily determine where the ball will fall because out of its initial speed, which is rather slow. As the game progresses, the ball will start to accelerate but the game slows down automatically every 100 points that the player earns.
Back to the Platinum Nintendo users, I would like to go into more detail on one issue.

In November 2009, Nintendo Japan announced that members of the Nintendo Club would be entitled to receive Ball as a prize. The unit looks almost identical to the original, although it features the Club Nintendo logo and a switch that allows the user to increase or decrease the volume. This award was only available to members of the Nintendo Club in Japan and was sent in April 2010 to those who earned enough points to achieve Platinum status. Unlike Nintendo America, Nintendo Japan automatically sent these units to eligible players without the user having to register online. Ball was later released on Nintendo America's Club Nintendo service, making it available to fans in the United States. At its release, more coins were needed than any other product: 1200. Let us rejoice that at least one preinstalled battery was supplied with each unit.

For specs lovers, here is a roundup of technical specifications of the Ball 2009 remake:

:: Size
Width:95mm, Height:63mm, Depth:11.5mm

:: Weight
51 grams (with batteries)

:: Battery
Two batteries (LR43 or SR43)

:: Battery life
With the watch on, the battery lasts about 6 months with LR43 batteries and 12 months with SR43 batteries. If the user plays only one hour a day, the system will last for six months with LR43 (the battery life with SR43 has not been released). The user can notice when the battery life is exhausted : the display fades and it becomes difficult to see the graphics.

:: Input
Left button (controls the character's arms and clock times),
Right button (check the character's arm as well as the clock minutes),
Game A button,
Game B button,
Time button (displays time),
ACL (allows the user to change the displayed time).

:: Temperature
Ball works correctly at temperatures between 50 ° and 104 ° Fahrenheit (10 ° -40 ° Celsius).

:: Precision of the watch
The average daily difference is estimated at 3 seconds at appropriate temperatures (see above).

:: Model number
AC-01.

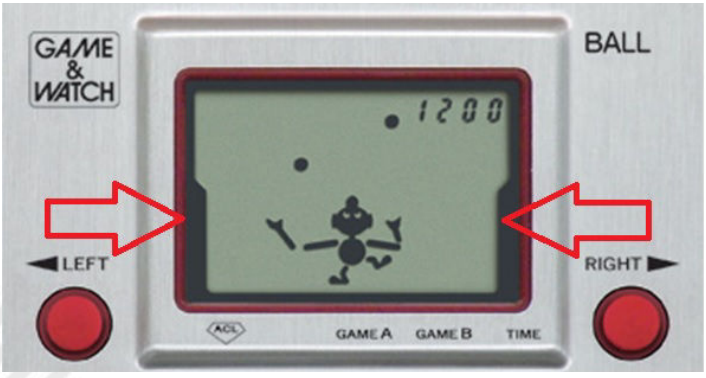


Figura 3

Figura 4

:: Price
Better not talk about it, a thorny issue. (see Figure 4)

That's all for today, dear readers!

I wanted to also talk about another beautiful project, but unfortunately in Covid19 time it was temporarily suspended. That's too bad! Anyway, let's hope I will manage to start working on it again soon. Ball is reborn after a third century, with great commitment we will see the light again too, united in a huge virtual embrace.

The next time, I hope to be able to talk about this second project, but given the unfortunately difficult times, I will most likely end up telling you about another recent, really adorable remake of the Game & Watch series (see Figure 5).

See you soon!

Figura 5

How I discovered RPG games on my TI99/4A

Or, how the monkey found the path of evolution...

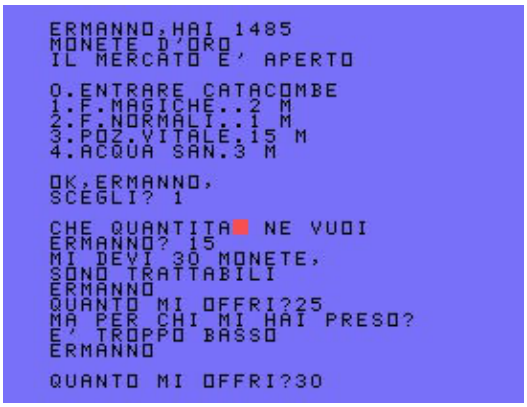
by Ermanno Betori
English translation by Cercamon – Revision by Baz

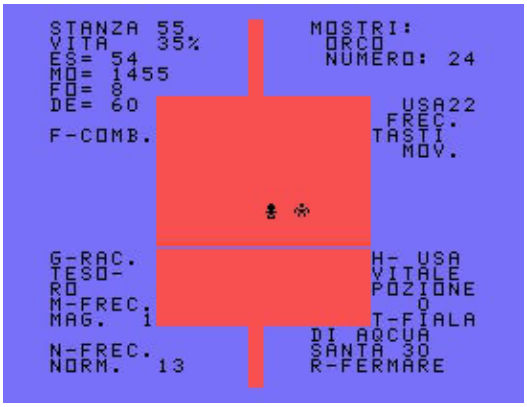
In my articles within previous issues, I described two Role Play Games for the MSX computer. In this one, I will describe my initial path down the road to enjoying the pleasure of playing RPGs by telling you about the games that were released for the TI99/4A that really made me see the light. I will start by telling my personal experience with home computers: going back to 1983, my relationship with computers can be summed up by this picture... (well, I am the humanoid on the right).

So I did what everyone did, i.e. looked out for help/hints and tips and software of course (mostly games), possibly for free, but... oops, there's a small space-time issue here. Back in 1983 there was no Internet, no BBSes, and no other form of quick contact with other computer fans, so really the only form of communication was through magazines!

Yes, those holy magazines! They were the only good source of computer knowledge, in the form of pages of BASIC listings. So I started learning about my computer from the bottom up, very slowly, by typing out the BASIC programs like a disciplined typist.

After discovering that a very long program I typed-in actually worked fine, I then found out that it didn't stay in memory when you turned the computer off (yes, I know, finding out about how the RAM memory works the HARD WAY was a rather painful and traumatic thing!). I then went on to the next step: the purchasing of a 'mass' memory device. So my new tape recorder gave me the confidence that all the programs I typed would not be "lost in time, like tears in rain".





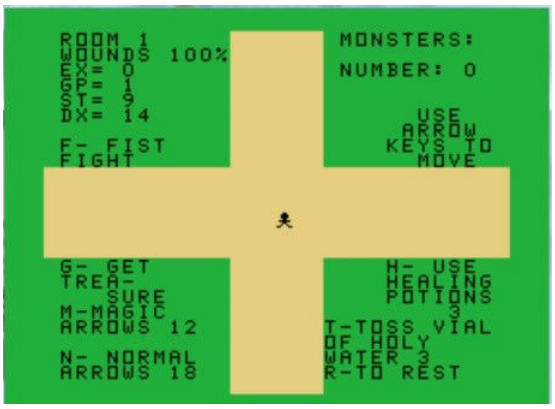
After losing (almost immediately!) several times playing "Catacombs", I finally had an idea about the basics of every RPG or adventure game:

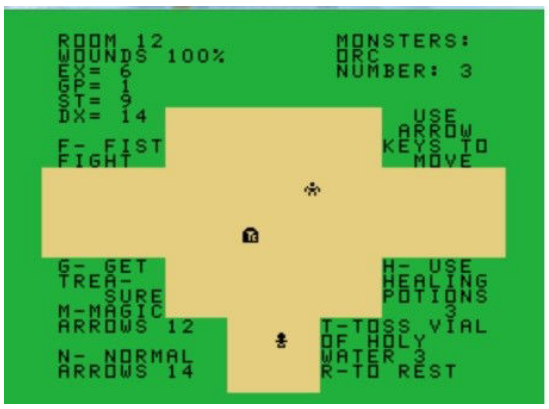
1. Never go around unprepared!
2. Merchants are usually there, so you better buy those damn arrows!
3. Death and Ghosts do not like Holy Water at all!

Now, technically speaking, the games were written in TI-BASIC, the default programming language available on the TI99/4A. It was of course very slow and lacking many features like sprites management. Yet even with these limitations, this particular game had all the basic elements of a good RPG: there was the hero, the monsters and the treasure. Yes! No princess to save... (bye-bye, Mario Bros.), there were only shoddy coins to find and a slight form of interaction with the dealer when purchasing weapons, etc.

Giving the game a score is not an easy task as it was my first RPG game experience ever, so I'll give it 100% for nostalgia reasons because I grew up with this game, but cannot give it more than 30-40% overall when we consider it's written in TI-99 BASIC.

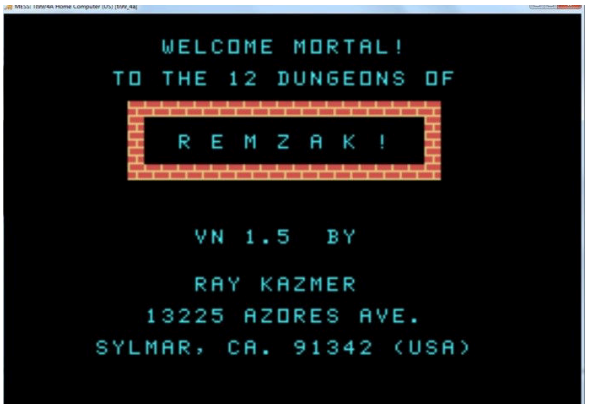
Years later, I discovered that this game "Catacombs" was published in one of the most famous Italian magazines of the time. It was a rough and ready version of a game called "Quest", originally published in a U.S. magazine. A sample of its bad translation in Italian is the word "Ghoul" which became "Morte" (Death). Excluding very few people who know about demonology (eg. priests, exorcists, etc.), in our native Italian horror and mystery stories, we only find ghosts, demons, phantoms and witches, but rarely, if ever, were there "ghouls", or vampires (demonic spirits that eat the bodies of the dead). This is an example of the translation mistakes that often populated the Italian RPG games and textual adventures. The screenshots on this page confirm extensively that in the 80s there was no concept of software copyright in Italy.

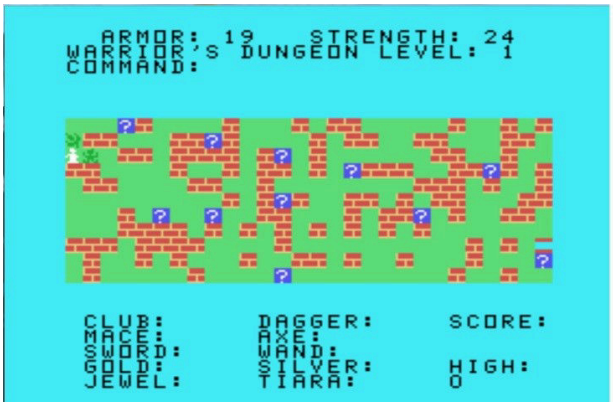




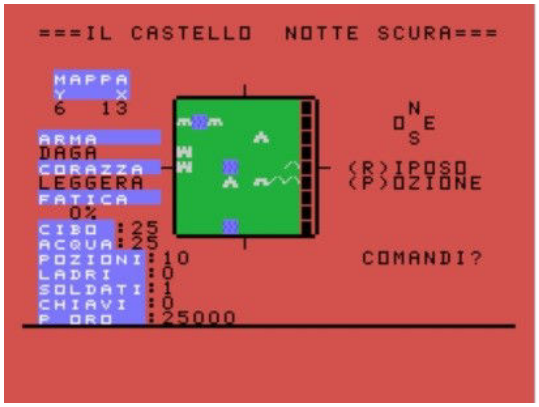
Other RPG games which were developed following along the lines of "Quest" were "Catacomb" and "Remzak" - both decent quality products but completely unknown in Italy. As a matter of fact, the big problem with the lack of software - especially in the first half of the 80s – was that very few exchanges took place with foreign computer clubs, groups or software houses, but this was destined to change in the next decade.

Screenshots from "Catacomb"





Screenshots from "Remzak"



However, both games feature the basic ingredients of a good RPG. "Remzak", in particular, offers a great variety of weapons, magical items and treasures to be found. It scores more or less the same as the previous title ("Quest"), let's say 50-60%, considering that these ones are technically more 'complex'.

When these games were first released in the States, I was still a youngster and had to grow up and learn more, as I was still tied to the (pirated) copy of "Quest"! But thanks to many brilliant Italian programmers since, I overcame my temporary delay in learning more about computers and entered the RPG world acquiring the "Novice" level.

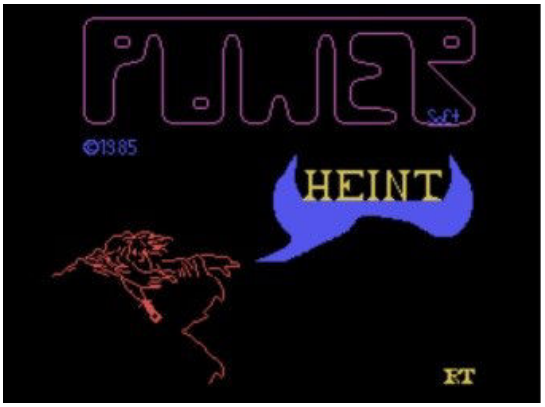
The TI-BASIC programs that gave me superior abilities were "Dark Night Castle" and "Monster Manor". The first title taught me how to survive in a maze full of traps and the other game how to survive in a simulated world. Indeed, "Dark Night Castle" featured the idea of showing the map of a closed world, with cities, hills, lakes and more. Further, the player could extensively interact with this world, building up new possibilities in a typical RPG. For example, wandering through the hills could easily lead to you finding giants in larger quantities than in any other places on the map.

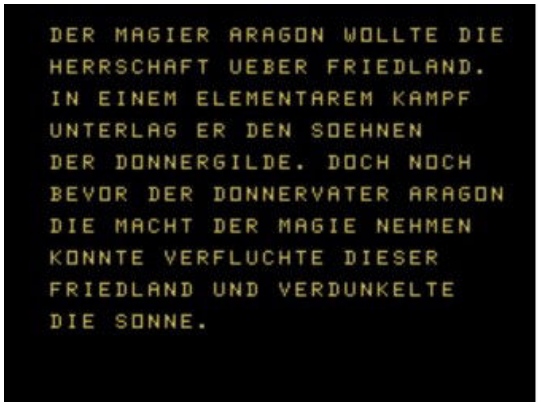
Hence, at last I was gaining experience in this type of game, and thanks to Extended Basic plus a memory expansion allowing for superior programming on the TI99/4A, I played several RPGs that enabled me to learn more quickly which gave me the chance to reach for advanced skills. I was finally entering into the 'history'

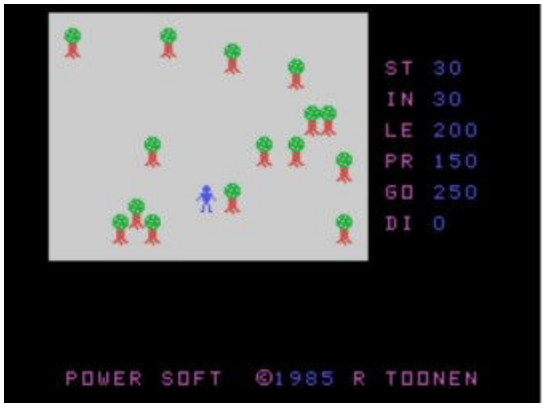
of the genre. Suddenly, new games opened up a whole new world to me: games like "Heint", "Old Dark Caves",

"Legends" and "Living Tomb". These four RPGs actually provided the fundamentals of the RPG games evolution.

Let's start off with "Heint". Programmed in X-Basic, this title contains the graphical innovation of using sprites and the hero does not automatically acquire new skills and levels. Actually his skills (strength, intelligence and stamina) increase as the game proceeds, but he has to pay for it! Does this ring a bell? No? Think about it... We will find this feature again in many modern role-playing games where the hero pays the guilds to be trained in strength, agility, sorcery, use of weapons and more (also think about the RPG "Thalion Amberstar" on Amiga/Atari ST). Here we have progress on the technical side but we are still far from the realms of excellence, so this title scores an overall 60-70%; not bad, considering it was programmed by a single coder.







Another good adventure/RPG game programmed in X-Basic was "Old Dark Caves" (it also has a sequel). This game offers a more arcade style of play, yet still features a typical RPG background. At the beginning of the game the player has the chance to select what to purchase in order to equip our hero. Indeed, you can select from different types of armour, weapons, spells and food and this actually affects the fighting and reaction abilities of our hero as well as our (poor) technique when fighting against several monsters. This game technically brings about further evolution via a different concept. As for the game's ideas, it is similar to "Hylide", but they unfortunately tend to remain 'unexpressed', so its score can't be more than 70-80%.

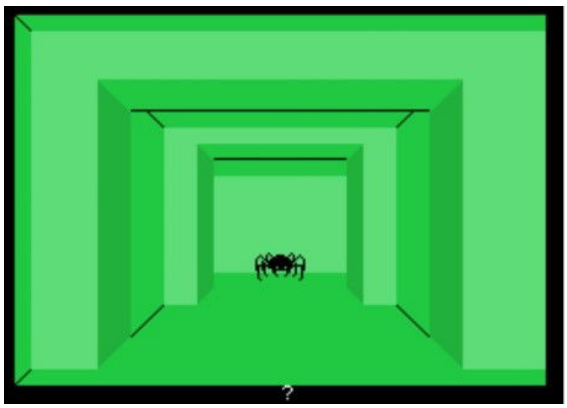
The last two games that I remember, with a greater degree of nostalgia, are "Legends" and "Living Tomb" because they ideally represent the 'transposition' on the TI99/4A of games that made history of the Role Playing Games. I'm talking about "Ultima" and "Dungeon Master", as they are still considered real masterpieces today.

What can I say about "Legends"?

THE PARTY IS NOW FACING 4 THUGS
YOUR OPTIONS, ERIC
1)HIT 2)LUNGE 3)PARRY 4)CAST
NAME HIT CON ARM MAG STA
1)ERIC 24 24 4 0 FIR
2)RAGNAR 12 12 3 12 RNG
3)WINTER 2 2 2 18 WIZ
4)GORT 16 16 2 16 CLE

It doesn't take too many words, it only takes a glance! "Legends" (it also had a sequel afterwards) is even today THE ultimate RPG for the TI99/4A. It features almost all the elements of the most famous "Ultima" series, including a high degree of difficulty and the supposed "ability to spiritually elevate" those who managed to complete it! This game offers the player the very top design of the RPGs in the 80s. The programmers of the Asgard label at the time really squeezed their brains out to obtain such wonderful graphics using a mixture of X-Basic and assembly routines. How about the score? Well, surely it goes up to 90-100%. It really is the best experience for any old-style RPG player.

Finally, let's end my trip through the RPGs on the TI99/4A by introducing "Living Tomb".



As previously mentioned, this title was an attempt to reproduce Dungeon Master's idea of creating a 3D-style

RPG, but it radically changed the interaction method making it less arcade and adding a more complex (and less intuitive) management system. This is one of the few 3D-style RPGs available for the TI99/4A. A fair score here can range between 80 and 90%, if we consider that this game, like "Heint", was designed and coded by a single programmer.

To conclude my journey through my own experiences with RPGs, I would like to thank the many programmers who created these fantastic games which opened my eyes to this wonderful world, indirectly increasing my curiosity about the cultural background this kind of game was based on. Basically, it's because of them that I bought and read and fully enjoyed books such as Tolkien's "Lord of the Rings".

Greetings also to all of the players who have at least once seen the hero of the RPGs in themselves!

Today the monkey that used to sit on my shoulder, no longer exists... It has been replaced by an old, wise adventurer who willingly awaits you at the inn for a friendly beer!

DISCLAIMER ABOUT THE ENGLISH VERSION OF RETROMAGAZINE

SOME OF THE FEATURED ARTICLES/CONTENTS WERE ORIGINALLY WRITTEN IN ENGLISH BY THEIR RESPECTIVE AUTHORS. THE OTHER ARTICLES HAVE BEEN AUTOMATICALLY PARSED BY MODERN TRANSLATION ENGINES AND THEN REVIEWED AND PROOF-READ BY MEMBERS OF THE EDITORIAL STAFF.

ERRATA CORRIGE FOR THE PREVIOUS ISSUE #00

JEAN-PAUL MARI - HELLO, I'M READING THE ENGLISH INTERNATIONAL VERSION ISSUE #0 AND SPOTTED A PROBLEM AT PAGE 6. THE TRS 80 DOESN'T USE 6809 BUT A Z80. NOTE: THE THOMSON FAMILY COMPUTERS (FROM FRANCE) USED A 6809.
RMW – THANK YOU SO MUCH, JEAN-PAUL. OF COURSE YOU'RE RIGHT, THE MOTOROLA 8-BIT 6809 WAS EMPLOYED IN THE THOMSON SERIES, WHICH WAS QUITE POPULAR IN FRANCE. SORRY FOR THE SMALL OVERSIGHT.

KEVIN KERFOOT - THERE'S AN ERROR IN THE ARTICLE ABOUT C64 DISK FORMATS ON PAGE 11: "THE 1541 DOS USES THE TRACK NUMBER 18 TO STORE THIS INFORMATION, THUS REDUCING THE AVAILABLE SPACE FOR USER DATA TO 169984 BYTES, CORRESPONDING TO 664 TRACKS (683-19) MULTIPLIED BY 256 BYTES." IT SHOULD SAY 664 SECTORS, THERE ARE ONLY 35 TRACKS ON A C64 DISKETTE.
RMW – THANK YOU KEVIN FOR BEING A CAREFUL READER! YES, IT SHOULD READ 664 SECTORS, NOT TRACKS. SORRY FOR THE TYPO.

KNIGHTMARE SAGA

Year: 1986
Publisher: Konami
Version: MSX
Genre: RPG

Talking about Knightmare (Majou Densetsu , Demon Castle Legend) is not easy at all.

It was produced by Konami Software in 1986 for the MSX standard only, making full use of the graphics capabilities of the TMS9918/29 video processor creating a perfect vertical scrolling and a wise use of the 16 colours available, while on the audio side it used the standard sound processor of the MSX1 models the General Instruments AY-3-8910 but with a surprising result. The soundtrack was written and executed by Miki Higashino, a musician who contributed to other games that became milestones such as Salamander, Gradius, Ye ar Kung fu, Suidoken 1 and 2, and so on.

In the game we are a nice blue Viking named "popolon" whose task is to save the sweet Princess Aphrodite, who has been locked in a castle by our worst enemy. To do this you have to cross 8 levels full of monsters, skeletons, and bats, with the boss at the end of each level as the icing on the cake.

Technically, the game is nothing more than an extremely smooth vertical scrolling shot'em up. The sprites of our enemies have been designed with extreme care to say the least and the difficulty is perfectly calibrated, although I have to admit that it is really hard to go beyond the fourth level. The programmers added the possibility to find bonuses that allowed us to change weapons, become invisible, stop time, etc. These appeared breaking grey stone slabs scattered everywhere. Some of them are hidden and appear out of nowhere when shooting them.

The game was so successful that the following year Konami produced two sequels: "THE MAZE OF GALIOUS" and "SHALOM-KNIGHTMARE 3 FINAL BATTLE".

The official versions of Knightmare to date are the one for MSX in 1986, the Korean freeware version made by independent programmer Kim Sungsik in 1990 for PC MS-Dos (the first Korean game to support EGA graphics), and the remakes for MS-DOS in 1992 (made by Friends Software, a group of hackers who also made a version of Ye ar Kung fu for PC in 2000), Wii (2009), WiiU (2014) and many mobile phones in 2003.

The Gold edition for MSX2 was released in 2006. In 2012, the ZX Spectrum version was published by climacus, who squeezed every single bit out of the ZX (https://www.youtube.com/watch?v=MlOxC3Xjelg) and finally the remake in 2014 of Nerlaska Studio (see the video from YouTube. A mashup with a beautiful soundtrack on electric guitar http://it.youtube.com/watch?v=vV2v3cpAnE).

Some facts about the GOLD version:
This version was born thanks to the hard work of Daniel J. Caetano and his Brazilian programmer friends from the software house Amusement Factory.

This group of Brazilian enthusiasts has thought of everything, in fact the GOLD version of Knightmare was released in CD-ROM format, they have revised and improved the code adapting it to the DOS of the MSX2, they have thought of the packaging, the user manual, the graphics of the CD, etc. In short, a great job.

The game contains all the levels of the original but with much better graphics and additional screens (like the initial one), and even the soundtrack has been totally rewritten and remixed with fantastic results. From the official website you can download all the tracks in MP3

version, the PDF manual, the CD cover and the entire game for MSX2 (http://www.caetano.eng.br/MSXPage/kmg/?p=5&l=en) PS: the site has also been translated into English.

Some clones that we will cover in detail in the next articles are:
Leonidas by Casio (see the video: https://www.youtube.com/watch?v=iw2DkKWiVSI) and Cyborg-Z and WON-SI-IN by Zemina (which translated means "prehistoric man"). The latter is really difficult. All have been produced for MSX computers.

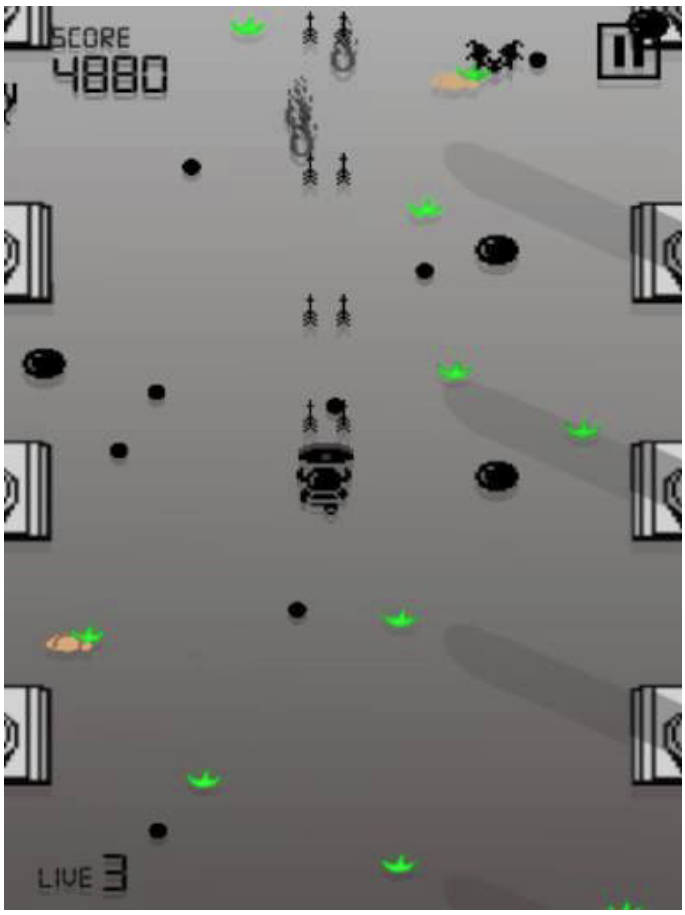
Another game clearly inspired by Knightmare was King's Knight Square (1989) for the NES (https://www.youtube.com/watch?v=D7MbzoSqccI)
We have yet another historical clone and it is the most shamelessly similar to Knightmare, on the verge of plagiarism, its name was THREE DRAGON STORY for Master System (Zemina, 1989). Take a look https://www.youtube.com/watch?v=zovFVa-YWsk

KNIGHTMARE REMAKE:
The remake of Knightmare was produced by a Spanish group called NERLASKA STUDIO, which is involved in the production of video games for PC and Smartphone.
The game bears the name of KNIGHTMARE TRIBUTE and is a true spectacle, very respectful of the original game. It has incredible graphics and fluidity but the strong point is the playability, practically the same present in MSX computers.

Nerlaska also made a game for iPhone, iPad and Android called GAZZEL QUEST. This game is also a Knightmare clone very well made, so Nerlaska should enter the Guinnes for being the only software house that made a remake and a clone of the same game.

Now Nerlaska officially no longer deals with indie video games, but from its ashes came http://demonvideogames.blogspot.com/ where you can find these games and others developed later on sale at bargain prices.

A fun fact is that before the official release of the remake by Nerlaska (now DemonVideoGames), there were countless remake attempts, lots of Beta versions and a huge amount of amateurial attempts, but never a real official version.



One of the most successful remakes was called IMPETUTH; released in Japan on December 29th, 2008 by the software house O-Games, the game has on only 5 levels, but with an impressive soundtrack, composed by Shinji Hosoe, who had previously worked for Namco on games such as Dragon Spirit and Ridge Racer. We can still admire a complete video on YouTube, the game is really beautiful. In an unprecedented way, it gives up the typical comic style of the MSX game, with a more attractive and serious appearance, with clear manga influences.

Brain Games (a group of Spanish MSX computer enthusiasts who created remakes of all the best MSX games, from Goonies to Maze of Galious) started the project of a 3D remake of Knightmare, but it was sadly never completed.

IMPETUTH VIDEO:
https://www.youtube.com/watch?v=UAqPtMDAYuo

3D KNIGHTMARE VIDEO:
https://www.youtube.com/watch?v=lX4b5C2YrY4

That's it for now. In the next articles I will continue to describe the games that were born after this forefather and that became part of the videogames history.

by **Ermanno Betori** and **Alessandro Tiso**

CITY BOMBER 4K

Publication Date: 2020
Publisher: Megastyle
Platform: Commodore 64
Version: Commodore 64
Genre: Action

Well, those of you who started its video game career with Commodore Vic 20 can't help but remember the simple but hypnotic arcade game where, on board a World War II plane, you had to clean up an entire city to ensure you could land safely without crashing into the buildings.

It was the famous BLITZ, one of the first shooters for the small jewel of Commodore, also known as "skyscrapers" in the compilations on cassette available in the newsstand in those years.

The plane, now without fuel, could do nothing but gradually lower itself at every step on the screen, with the luck of being able to drop infinite bombs through which to demolish entire buildings.

You couldn't slow down or turn, but you could just drop bombs, obviously, without any electronic tool to tell us the right time. However, the bombs could not be fired in quick succession, the player had to wait for the detonation of the just dropped one before he could release another...

In order to play City Bomber 4K, however, we will need a Commodore 64 or a free emulator such as the VICE and, besides being very fun, it was also very enjoyable from a graphical/sound point of view, given its weight of only 4K.

The game can be downloaded free of charge in d64 format and is also available in physical version on cartridge along with several other games by different authors all playable by pressing a single button.

Free digital versoin:
https://megastyle.itch.io/citybomber-4k
Single Button Games – Physical version on cartridge:
https://shorturl.at/hJMW2

by **Flavio Soldani**

Blitz - original VIC20 version

City Bomber 4K - C64 version

OUR FINAL SCORE

» Gameplay 89%

A title for Commodore VIC 20, very well revisited in 2020 for Commodore 64, to prove that true classics never die.
The game also involves setting the difficulty so as to make it suitable for all palates.

» Longevity 90%

Obviously suitable for short and intense games like most good arcades in the 1980s. However, if the fun of playing Blitz has remained unchanged in all these years (the original is from 1982) I would say that the longevity test has been largely passed.

CLOUDY MOUNTAIN - A boardgame for one player

Hot Topic!!!

by Starfox Mulder

Cloudy Mountain is a solo board game inspired by one of my favorite video games of all time (Advanced Dungeons & Dragons for Mattel Intellivision).
You only need 8 six-sided dice, a printer, a sheet and a pencil. If you had colored pencils it would be better, but it's not necessary.

Character Tab
In this game you play an elf archer on his quest for the King's crown. You will start the game with 6 hit points, a quiver with 6 arrows and a 6-day supply of food. To show this, place 3 dice, with the top face showing the number 6, in the appropriate spaces.

- Hit Points: while exploring dungeons you may encounter different types of monsters that will be able to hurt you. If this happens, change the die to show the new number. When you reach 0 hit points the game is over and you lose.

- Quiver and arrows: you start the game with 6 arrows and you will use them in the dungeons fighting monsters. Each time you change the die to reflect the actual number of arrows you have. In the dungeons you may find a new quiver (never more than one per dungeon) and this will be indicated with an additional die placed next to the previous one, with the top face showing the number 6. In the inventory we have maximum space for 3 quivers. If we find more we will simply refill the owned ones as much as possible.

- Food supplies: every step we take on the game map forces us to consume at least one food resource, changing the die to keep track of them. In the dungeons we could find additional food supplies, which allows us to refill the inventory back to maximum (never more than 6). Moving into a dungeon does not consume food supplies while outside, in case we run out of food supplies, we will be forced to hunt. Hunting makes us use 1 arrow for each food resource we get, which means we will consume 1 arrow for each movement without food supplies.

To sum up:
Whenever 1 food resource is required and you do not have it, you use 1 arrow.
Whenever 1 arrow is required but you do not have it, you spend 1 hit point.

Explore the map
Some parts of the map are already revealed and are the same in every game. The player starts at his place (left side of the sheet) and must reach the foggy mountain (right side). Yellow tiles represent plains, and entering them involves the use of a food resource per tile.
The blue tiles are rivers and can only be entered once the boat is found inside a blue dungeon.
The “gate” tiles cannot be entered unless you have the key (which can be found in purple dungeons). They can be entered only from the two open sides.

The biggest part of the game is created on the fly by the player himself when his character moves to a tile bordering unexplored areas.
Moving to such a tile forces the player to explore them immediately, rolling 2 dice for each box whose content you want to reveal.

One more roll is needed if you find a dungeon.
The table below lists the results of the roll and what you will need to put in the revealed tile (use the colours you have, you are clearly free to draw forests or mountains if you can).

Result 2D6	Map of the World
2-6	Plain
7	Forest
8	Mountain
9-12	Dungeon

Result 2D6	Dungeon
2-6	White
7-8	Blu
9-10	Red
11-12	Purple

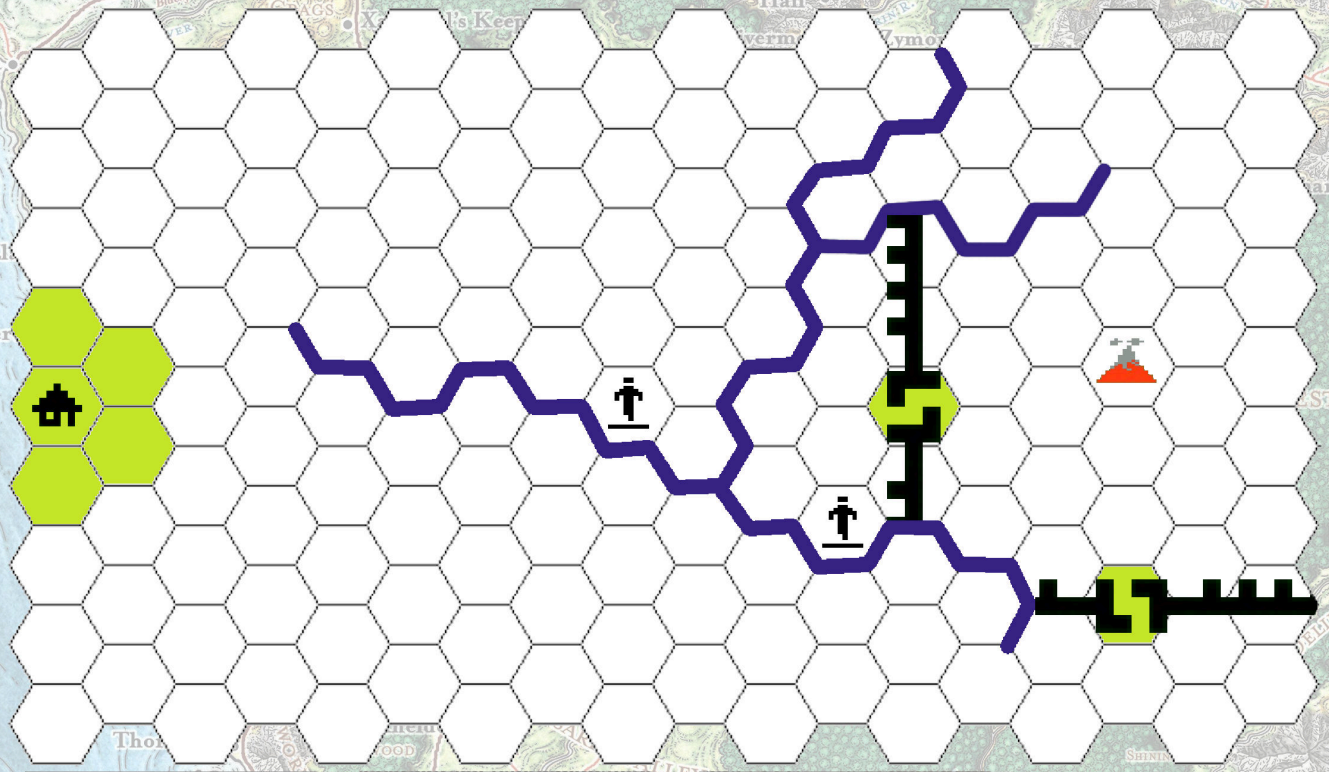
- Plains : Write a P with the pencil or colour the tile yellow. Plains can be entered without risk, consuming a food resource.

- Forest: Write an F or colour the box green. Forests can be crossed consuming a food resource if you have an axe. If you don’t, you need to spend one hit point and two food resources per tile.

- Mountain: Write an M or colour the tile brown. Mountains are impervious and very dangerous places that you would better avoid. If not possible, they can only be crossed consuming one hit point and three food resources.

- Dungeon: Draw a triangle and inside add the letter B (Blue), R (Red), P (Purple) or leave it empty in case it is white. If possible, it would be preferable to colour it. Entering a dungeon consumes a food resource.

Cloudy Mountain



2 - 6	L	2 - 6
7	F	7 - 8
8	M	9 - 10
9 - 12		11 - 12

= - 1
= -
= -
= -
= - 1 -
= + 1
= Full
= No injury from the next
= + 1
= Exit
= - 1 -

Health Points

Dungeon Room

Food

L = -1
F = -2 -1
M = -3 -1
= -1

cross the river

forests as lawlands

open the gate

= + 1
= -
= -
= - 1

DUNGEONS

A Dungeon is made of 6 rooms generated procedurally on the fly rolling 2d6. To keep track of which room you are exploring use a dice and turn it every time you move.
Facing an enemy does not require rolling more dice. Instead check the previous roll on the following tables.

Dice score	Encounter/ Event	Conseguencies/ Damage
Double 1	Monster: Abomination	-1 point injury
Double 2	Monster: Bat	Discard tracks (if you found them)
Double 3	Monster: Rat	-1 arrow
Double 4	Monster: Spider	-2 arrows
Double 5	Monster: Boss	-1 point injury; -2 arrows; treasure
Double 6	healing potion	+1 point injury
Offset of 1 between scores	Food supply	Refill your supplies
Offset of 2 between scores	Tracks	No injury until the next Boss
Offset of 3 between scores	Quiver	+1 quiver
Offset of 4 between scores	Exit	you found the exit
Offset of 5 between scores	Monster: Boss	-1 point injury; -2 arrows; treasure

Caveat: If you get the same result in more than one room, the repeated ones are considered empty.
Exit: in each dungeon there is an exit and if you do not find it in the first five rooms it will be in the sixth. Finding the exit doesn't necessarily mean you want to use it, you can leave it for later and continue exploring the remaining rooms.
Boss and treasure: in each dungeon there is a boss and if you do not meet him in the first four rooms he will be in the fifth. Boss and treasure type are however different depending on the colour of the Dungeon

Dungeon's color	Boss	Treasure
White	No changes	Quiver (No more than one for dungeon)
Blu	1more arrow	Boat (allow you to cross river)
Red	1 more arrow	Axe (allow you to cross the forestes)
Purple	1 more arrow, 1 more hit point	Key (allows you to go through the gates)

THE MERCHANT

There are two merchant tiles on the standard game map. In terms of movements they are considered plains, and when you enter them you may choose to sell one or more of the resources you collected in the dungeons in exchange for a healing potion and regain one or more lost hit points. The resources you can sell are: a quiver (with 6 arrows, not less), the boat, the axe and the key.

THE FOGGY MOUNTAIN

The foggy mountain is actually a dungeon but it contains twice the rooms (when you reach the sixth, start counting again with the die) and makes you face two bosses to collect the two parts the crown has been divided into. Bosses are two-headed dragons and require 4 arrows each to be defeated, and also inflict 2 wounds.
The rule that identical results in the discovery of rooms are calculated as an empty room does not apply in the foggy mountain. This is why you could, for example, find the tracks of monsters twice and get out unharmed from the next two battles.
If bosses don't show up in the previous rooms, the eleventh and twelfth contain them.

VICTORY AND DEFEAT

To win the game you need to reach the foggy mountain and defeat the two bosses, thus collecting the two parts of the King's crown. Once that's done, the game ends and you triumph.
Any time your hit points reach 0, the game ends and you are defeated.

TIPS ON HOW TO USE THIS GAME:

- Print the game map in colour and place it in a clear plastic envelope, on top of which you can write freely with water-based markers and erase at the end of the game.
- If you don't have colours, print the game map in black and white and write on it with a pen, pencil or coloured pencils.
- Use your pen or a black marker to record your hero's path on the map, so you remember where he has already gone. If you return to a dungeon that you have already explored this will be considered plains.
- You can find food supplies necessary for survival only in dungeons, which is why my advice is to explore as many as you can.

GRAND THEFT AUTO I

Year: 1997
Developer: DMA Design
Publisher: BMG Interactive
Platform: MS DOS
Genre: Action/Adventure

In this new series of articles, we're going to review those video games that don't need any introduction.
Actually, at least once in their life each of us has experienced what it's like to get excited playing a title in this series by Rockstar Games. But when it comes to it the world of video game players (and not only) splits in half.

There are those like me who spent their childhood and youth (and still do it today) running through the streets of Liberty City, Vice City, or the immense state of San Andreas, making this video game series a very important way to let off steam during our lifetime. Others, on the other hand, have always hated these video games because of their gratuitous violence against unsuspecting citizens.

It is well known, that some players of these games have committed murders, claiming to be fans of this series (including Manhunt). Obviously, however, this is and will always be an excuse to justify such an action. Well, yes... This is GTA we're talking about!

"Grand Theft Auto" is the name of the first title in this series, released by DMA Design (now Rockstar North) on December 14, 1997 for MS-DOS, PlayStation 1 and Game Boy. There are three settings (Liberty City, Vice City and San Andreas), and we will play the role of a criminal who is recruited by various local bosses to carry out missions and jobs. However, nothing prevents us from wandering freely around the three cities, although aerial views are certainly not the best.

Grand Theft Auto is a game full of fun facts. Among them it definitely stands out the fact that we can find in the sequels the names of the cities in which it is set. One of the bosses will also be found in the masterpiece GTA 3, El Burro.

We will also find the beloved Pay

Fansites:
GTA Expert: www.gta-expert.it
GTA Series: www.gta-series.com
GTA Vision: www.gtavision.com
GTA Forum: www.gtaforums.com

'n' Spray that help us a lot during a chase, as they make it possible to repaint our car. And it doesn't end here… Did you know that Head Radio is appearing in this very first chapter of GTA? And that a heavily aged Donald Love will be found right here?

Unlike modern GTAs, to complete a mission in Grand Theft Auto you need to get a certain score, usable as money. The weapons available in the game are more or less the ones we still use today, including guns, machine guns, flamethrowers and rocket launchers.

The game mechanics, on the other hand, are the same used for example in GTA V, i.e. the presence of the police. As long as we're good citizens, no cop will be on our tail. As we begin to break the law, we will find ourselves wandering the streets overrun by policemen and then by the military if our level of suspicion reaches 5 or 6 stars.

One of the great flaws (today we would say this) of the video game is certainly the very poor graphics with so much view from above (aerial view), but on the other hand what could be expected from a video game released more than 20 years ago. There was not even a map with which you could find you bearings in the cities.

Grand Theft Auto has been released several times as a free download by Rockstar Games but the biggest news is that a dedicated page was created on the most famous video game store...I mean Steam! On this platform there is this game, followed by GTA 2 and GTA London... Isn't Steam maybe considering selling these timeless titles?
If so, the first downloaded copy will undoubtedly be mine ;) It is impossible to miss such an opportunity. As I said in the article on RM22 of the Atari 80 Games Collection, owning these video games at the time meant nothing. Today, however, they have invaluable value… probably not in monetary terms, but surely from the point of view of affection!

If you have a copy, then take the opportunity to reinstall it on your computers. Maybe Windows 98 emulation in Virtualbox (article on RM22) in case you don't have a pretty old computer.

I'm sure with these first three articles I wrote I made all the RetroMagazine readers feel a lot of nostalgia. I hope you can live hours and hours of endless fun and memories playing with these video games even installed on an old operating system.

by **Marco Fiaschi**

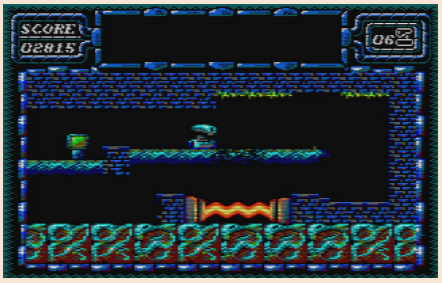
OUR FINAL SCORE

» Gameplay 60%
This first chapter of the series involves the use of an aerial view, which makes movement within Liberty City quite complex. In the PC version this made it even more difficult to search the surrounding environment. For this reason I give the game a lukewarm 6. As far as graphics and gaming mechanics are concerned, nothing to say. We're still talking about a '97 video game!

» Longevity 80%
Grand Theft Auto I, according to the purists of the video game series, represents a pillar with such a powerful foundation that it would be almost impossible today to create today a "more contemporary" product that uses the same original mechanics. Not surprisingly, in many video game top tens of the time, GTA I was always among the best, usually between the second and the third place (highly deserved in my opinion), overtaken only by GTA 3 and GTA Vice City. Considering that Rockstar Games is still very fond of this title, as probably many of us are, a score of 8/10 is more than deserved!

TRAKERS

Developer: Gary e Scott Kennedy
Year: 1991
Version: Amstrad CPC
Genre: Esploratory/Puzzle



In 1986, a film called 'Short Circuit' was released at the box office, starring a robot with semi-humanoid features. The film was fairly successful in Italy and around the world thanks to the main character and its friendly behaviour.

The plot of the film is simple, but at the same time intriguing; during a performance, one of the 5 robots created by NOVA Robotics, due to a short circuit generated by lightning, finds himself wandering free in the world and slowly acquires self-awareness...

Does Trakers' robot only remind me of Number Five, the star of Short Circuit? Probably not, because the story is somehow similar, too.
During an exploratory mission, Trakers, this is the name of the protagonist of our game, due to a misunderstanding, is forgotten by the crew of the Starl spaceship, alone on the planet Magellanx.

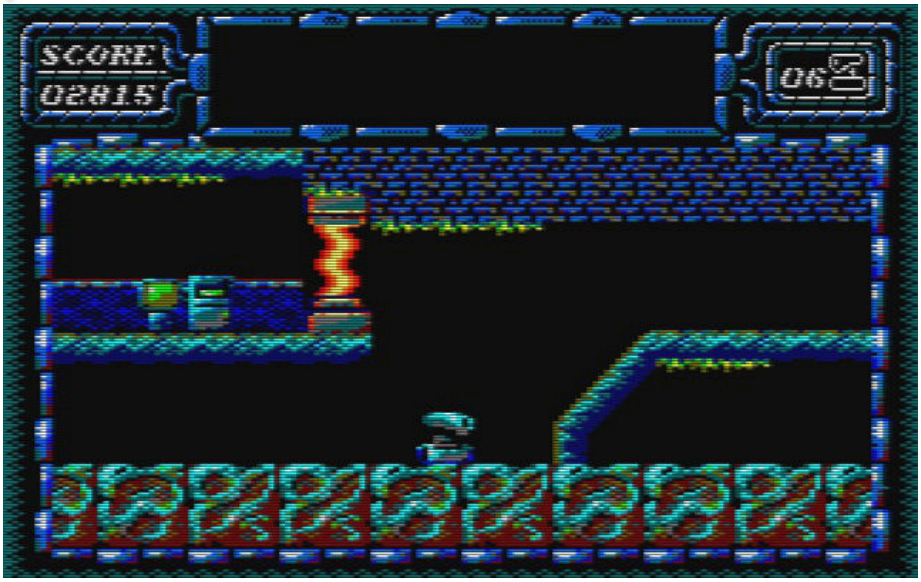
Trakers, just like Number 5, is equipped with tracks which allow him to move on any terrain and make long jumps to avoid any obstacles. Trakers' luck is that Magellanx has an auotomatic system that can bring him back to the Starl spaceship, but, to activate it, he needs at least 10,000 credits! OMG, even on a lost planet on the edge of the galaxy, money makes a difference...

So our robot goes looking to collect the money to reunite with his friends. I would have stayed on Magellanx to have a good life, but Trakers seems to be of a different opinion and so our job is to help him in this endeavor.

The game
Trakers is a fixed-screen exploration game, where we need to collect objects to solve puzzles and continue on the adventure.

Obviously, the exploratory path is made more difficult by the presence of several aliens whose only touch is lethal to Trakers and the planet itself is filled with quicksand, corrosive sewage and several other traps. Looks like that Trakers doesn't have an easy life on Magellanx... Fortunately, however, he has an initial supply of 9 lives with which he can face all dangers with relative safety.

Overall, the game is eye-catching, the graphic is not superlative, but it



is detailed enough to catch the player to see what awaits him on the next screen.

The colors are used wisely in different settings, interior, exterior, caves... And the effect is very realistic.

Graphically speaking the game has also a peculiarity: the area around the score and the messages is made in MODE 1, the actual game area, the one where Trakers moves, is IN MODE 0.

The sound is completely absent, except for some special effects present only here and there in some spot locations. Personally, I appreciate it!

After the first easily solvable puzzles and having visited the first screens without losing any lives, you will end up losing several lives just to figure out how to deal with a singular step. In addition, after the first easily identifiable objects, you will find yourself with tools in your hands that you don't knowhow to use. It will take you some time to figure out what to use and where, but you'll never feel that sense of frustration that would make you stop playing.

Conclusions
If you are a fan of exploration games where you have to collect objects, figure out where to use them and you are not afraid to visit the same screens more than once, you will find Trakers very attractive. If you prefer to spoil for a fight or blast aliens and everything that moves, stay away from it, there is no violence in here.

Homebrew
Trakers was born in 1991 as a homebrew game created by brothers Gary and Scott Kennedy. Initially the two had attracted interest from some software houses, but in the end they decided to sell the game directly, without intermediaries, to maximize profits. Given the quality of the game, I don't think they had much trouble selling several copies.

Where to find it
The game can be downloaded from here:
https://www.cpc-power.com/index.php?page=detail&onglet=pub&num=22

How to proceed in the game
To proceed in the game you need a good memory or draw a map!
How about to make one and send it to us to help the other players?
We will gladly publish it, as we did with the Abbaye des Morts.

Do you need any help?
If you are stuck somewhere, you can find a Trakers longplay here:
https://www.youtube.com/watch?v=Z2K3Nur0yy4

More than an hour of play! You'll be busy for a while.

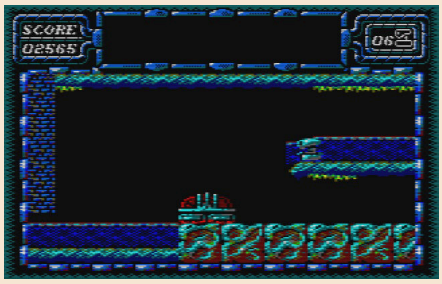
Have fun!

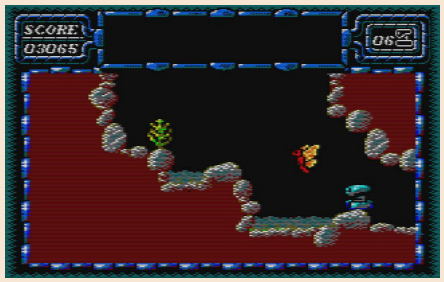
by **Francesco Fiorentini**

OUR FINAL SCORE

» Gameplay 80%
An honest game, well done from a technical point of view. If you love exploration games, you'll love it.

» Longevity 90%
Initially simple, the difficulty increases after the first 2 or 3 puzzles... More than 100 screens to visit and about an actual hour of play needed to complete it. Is there anything else you need?





CRYSTAL CASTLES

Publisher: Psytronik
Year: 2020
Version: MS DOS
Genre: Action/Adventure

Talking about collecting, nowadays, has become a “ghost" activity. I would ask any of you readers to know if you too were ever taken for a fool when you started collecting items from your past, or even the period before your birth.

For some years now, I have also started collecting, especially music CDs and computer video games. Luckily, it's not my case what reported a few lines up, but it was close... In some people's eyes, we look like crazy guys, because they think it's not normal to collect old objects. But this is where they're wrong. These articles that I am writing (and that I hope to write a hundred more times in the future) for RetroMagazine, are largely about my past.

And I certainly couldn't miss the article on Crystal Castles' much-loved Bentley Bear. Here, let's take a look at this very video game...

A lot of people will probably think, "But this 22-year-old guy still plays with Bentley? ” and many other similar things. Well, the answer is YES.

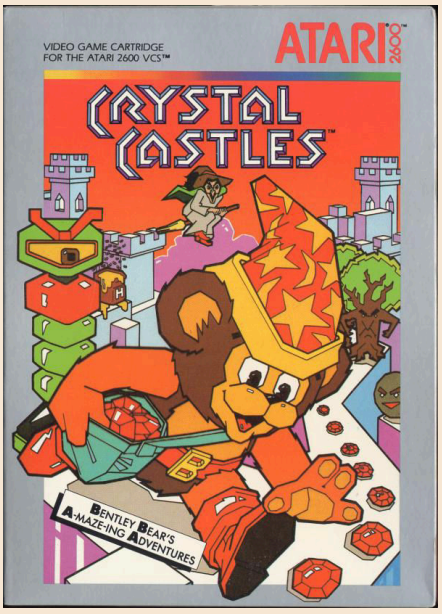
I believe that for us collectors, it is an increasingly strong emotion to be able to relive those instants, lived in particular moments of our lives. Among these, there is surely the adventure of Franz X Lanzinger. Of course I wasn't born when the Crystal Castles booth came out, but I was able to play it for the first time on the album Atari 80 Classic Games In One (see Article on italian RM22).

Left in a drawer for years, and found a few weeks ago, this record allowed me to relive the moments when I was just a carefree child and a relentless primordial gamer of computer games.

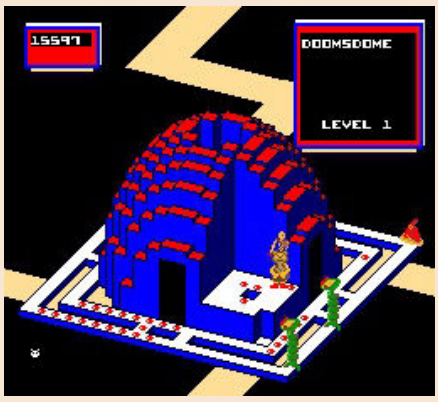
The adventure that struck me the most is the one of Bentley. The teddy bear who had to defeat the witch Berthilda by crossing 9 levels in 4 different castles, featured by the presence of an ever-increasing number of enemies and an ever-increasing difficulty. In the game it was possible to meet monsters of all kinds: Crystal Balls, or balls that roll on the playing field, eat the gems of Bentley and... even if you don't pay enough attention, witches, dancing skeletons, trees, swarms of wasps etc... I mean, there's nothing missing from the game!

Crystal Castles was also released for Atari 2600 and C64 but, for obvious reasons, was realized with very poor graphics, resulting in the presence of abstract geometric

figures in which we had to move to get the gems. The success of the arcade, awarded by the sale of 5380 cabins, was considered the best ever made by Atari. They had a trackball in the middle and the Start and Fire buttons on both sides.



Cover ATARI e cabinato



As for the curiosities of this game, we can find a lot of them. What certainly stands out most from the others is the particular shape of the structure of the castles where, if you look carefully, can notice that they are in fact the initials of the names and surnames of Lanzinger's friends.

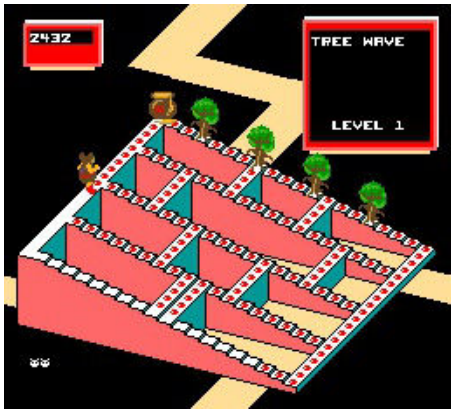
Over time, however, it became known that the initials actually referred to the names and surnames of the most legendary players in the game itself.

In 2012, the design of a side scrolling video game (Super Mario style) began with the characters of Crystal Castles, during which Bentley will have to recover 5 precious jewels stolen by the witch Berthilda.

Last but not least, we find curiosity about the legend of Crystal Castles, namely Frank Seay, the player who reached the world record of 910722 points in 1998, and no one ever managed to beat him again.

Maybe it's thanks to him that we know how the game actually ends. There are those who claimed that the game was actually infinite, and those who believed that the game was based on a single level that changed every time all the gems were collected. Actually, there was an end of the game.

After all that effort, we may have



expected a little more recognition, but it depends mainly on the number of lives available during the completion of level 10.

One thing is certain... the phrase “I GIVE UP: YOU WIN” is not taken away from us!

And getting this screen means going straight into Crystal Castles' Hall of Fame!

by **Marco Fiaschi**

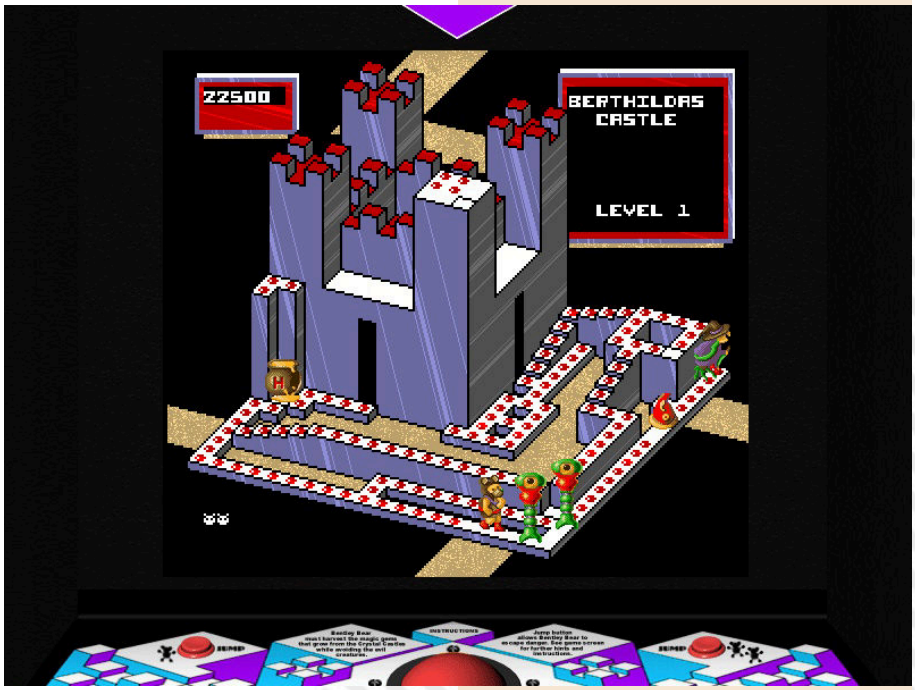
OUR FINAL SCORE

» Gameplay 70%

Please note that in this article we have relied on the PC version contained in the Atari 80 Games Collection In One (unfortunately we do not have a cabinet :)). You can proceed through the levels using only one mouse, which makes Bentley quite controllable. If we were to talk about the original version per cabin, it would not even be enough, due to the presence of an extremely sensitive and deteriorating trackball. Many players found Crystal Castles very difficult to play on the cabin.

» Longevity 60%

Even in terms of longevity Crystal Castles unfortunately does not excel. We are talking about a video game that has now been forgotten. Although the title was present in every Atari collection, unfortunately Crystal Castles did not enter in the 21st century in a triumphant way (although today it is still considered one of the best booths ever built). I invite all the players of RM to document themselves on this video game and give it a try. More than modern games! But beware, Bentley's adventures are addictive!



CORNER OF DARKNESS
SEPTERRA CORE

Developer: Valkyrie Studios
Year: 1999
Version: PC CD-ROM
Genre: RPG

Welcome back to the section dedicated to “peculiar” games. In this episode we will deal with Septerra Core, a rather unique title, born as a result of a long and difficult gestation. It all began in the studies of Rabid Entertainment, a software house formerly known as ICOM Simulations (authors of Shadowgate, the Dèjà Vu franchise and the Sherlock Holmes Consulting Detective series). Unlike previous titles, Septerra Core is a western RPG, although developed with a style inspired by the Japanese classics one can find on consoles.

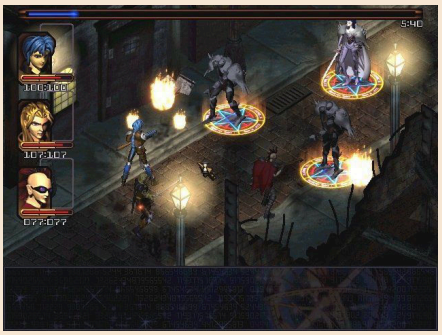
During development Rabid Entertainment ended up in financial difficulty and it was forced to close its doors. Part of the team then founded Valkyrie Studios and, thanks to the support of the legendary Monolith Productions, completed the game. Unfortunately, the title proved to be a commercial failure and caused the premature departure of the study.

Septerra Core is set on a uniquely shaped planet, with seven continents revolving around a central nucleus, connected together by a sort of organic backbone which regulates its movement. The Creator (the god who forged the world) left two keys to access the nucleus, which is activated every hundred years by sunlight filtering through a temporary passageway between continents. Legend has it that whoever is able to discover the secrets hidden within, will receive the Gift of the Creator and knowledge to enter the Kingdom of Heaven.

In the past a fallen angel tried to enter the nucleus, stealing the keys; the Creator was then forced to bring his son Marduk down to Septerra. After a long battle, the young man killed the demon with the help of an army composed of warriors from the seven continents, then decided to live on the planet and to give birth to the Line of the Chosen. Many centuries have passed since then, but the desire to discover the secret of the Core fascinates Marduk's descendants. Doskias, General of the Chosen Ones, decides to retrieve the keys and force the opening, unaware of the dangers it will bring to the whole world. In the shoes of Maya, a young orphan living in the city of Oasis, on the second continent (just below that of the Chosen Ones), we will have to try to defeat the tyrant, who we will also discover was the cause of our parents' death.

From a graphic standpoint, Septerra Core presents a vast and captivating gaming world, entirely made of pre-rendered two-dimensional graphics. The backdrops are extremely well cared for, rich in fixed and moving details. The characters are also two-dimensional and, although they are not animated in an excellent way, they do their part, also thanks to a good characterization of the avatars' faces during dialogues. The gaming world is enriched by a well-crafted system of dynamic lights and shadows, as well as a good compartment of effects during battles. Added to this is the presence of choreographic sequences in computer graphics, which highlight the salient moments of the story.

The audio sector uses a discrete quality soundtrack composed by Martin O’Donnell (responsible for the music of many Bungie house titles), which alternates well-crafted themes for the exploration phases of the main map with monotone compositions for battles. The sound effects are good, although they do not reach peaks of excellence. What is surprising, however, is the speech: each individual character has its personal voice-over, with a quality that varies from good to



excellent. This helps to have a more effective characterization and represents a great step forward compared to the competition of the time.

Septerra Core is a Japanese-style role-playing game that alternates stages of exploration, dialogue and puzzle solving, with turn-based combat sequences. We can say that, with all the differences, if you've played a classic episode of Final Fantasy, it won't take you long to settle in. There are all the classic styles of the genre, from equipment to magic, represented here by cards similar to tarot cards, which draw energy from the core of the planet and which can even be combined for more devastating effects.

The combat system is interesting and, despite the slowness of the battles (due to the impossibility of skipping animations), it is rather immediate and fun, thanks to the possibility of carrying three different types of attack, depending on the engaging time.

It is also interesting to note that enemies are always visible on screen during the exploration phase, as well as constantly on the move. This will allow us to avoid a few too many clashes if we want to. Another important feature is the presence of a real time updated map of the area where we are, which can be called at any time with the TAB key. The interface is not very comfortable, especially in the inventory and equipment management menu, but eventually you get used to it.

The title is entirely playable with the mouse, but if you prefer you can also use the keyboard to move your character directly. The game lenght is absolutely remarkable, ranging from 55 to 60 hours of play, which undoubtedly represents a good achievement for such a production. The level of difficulty is within average for this type of games, although it has a few peaks more complicated than expected. Luckily, you can save the game in any situation and this helps to avoid the sense of frustration.

Despite being available even nowadays on Steam and Good Old Games, Septerra Core remains largely unknown to the general public, although over time it has obtained a niche of faithful fans. If you like classic Japanese-style RPGs, with a solid narrative and a good aesthetic layout, this game may be just what you are looking for, as long as you know English, since there is no Italian version.

Also, don't forget to update (if you have the original CD) the title with patch 1.04, so as to avoid annoying bugs that could compromise the experience.

It's time to say goodbye, but remember our motto: "Stay Hungry, stay Obscure!".

See you next time!

by **Federico "Arzak1" Gori**

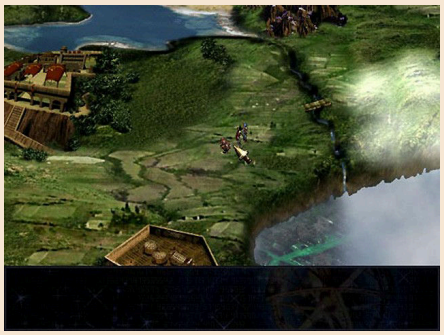
OUR FINAL SCORE

» Gameplay 80%

Inspired by classic Japanese RPGs, the game offers a successful alchemy between exploration and turn-based battles.

» Longevity 90%

The title is an absolutely lenghty ones, as per tradition of the genre. If you are ready to give it the time it deserves, a long and fullfilling experience awaits you.



SONIC THE HEDGEHOG

Editor: Sega
Developer: Sega Sonic team
Platform: Sega Megadrive, Sega Master System, Game Gear, Virtual console Nintendo
Reviewed version: Sega Megadrive
Genre: Platform

Is it still fun to run as Sonic, even after 29 years?

These years have passed quickly, from 1991 to 2020. Fast and fuzzy like Sega's blue hero. It's hard to believe that it's been so long since its release (looks like yesterday, E/N) since the hedgehog created by Yuji Naka and the Sonic Team first launched onto the video game scene.

Sonic the Hedgehog, also known as Sonic 1, is the first product where the mascot Sega made its debut. I arrived inside our megadrive/genesis in the hot June of 1991 accompanied by a very aggressive marketing campaign and creating a legendary rivalry with Mario of Nintendo opening the dances to the wars console of the 90s.

Its cool gameplay and, above all, its incredible sense of speed have made it a classic and a standard for platform games of the period.

Initially proposed for the 16bit Sega, it was immediately converted with a completely different version but really exciting also for the Master System and the Game Gear laptop. Later also on Gameboy Advace and the virtual consoles of Wii and WII U. It generated dozens of sequels and spin-offs, but the magic of the first three chapters and their innovativeness.

Sonic, the character, was born of necessity. Sega saw that Nintendo was killing with the NES and then with the Super Nintendo the market through the range of platforms starring Mario, and decided to replace his time mascots (who remembers Alex Kidd? NdR) with something different. A hero other than Mario, more dynamic, bolder and definitely faster.

As the team's development shifted to a game focused exclusively on speed, the candidate characters were reduced to a handful of animal designs; animals that could potentially run very fast and attack without losing their momentum. The team focused on two types of creatures that could roll their bodies into a ball: an armadillo and a hedgehog.

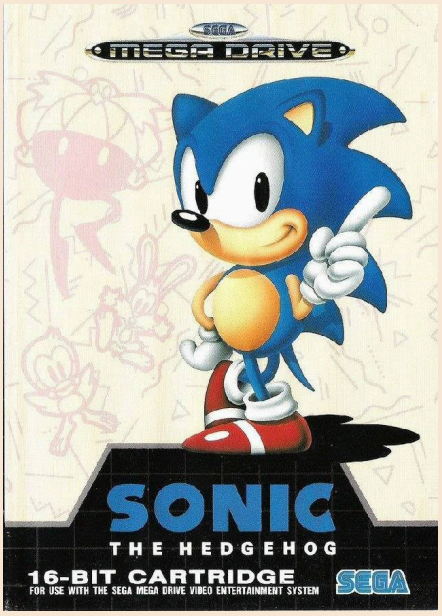
The idea was that the shape of the ball could serve as a rolling and jumping attachment without slowing the sense of speed of the game. The design of the armadillo was not completely abandoned, as it later became the basis of Sonic's universe character, Mighty the Armadillo. But the idea of the hedgehog won, giving rise to Sonic himself.

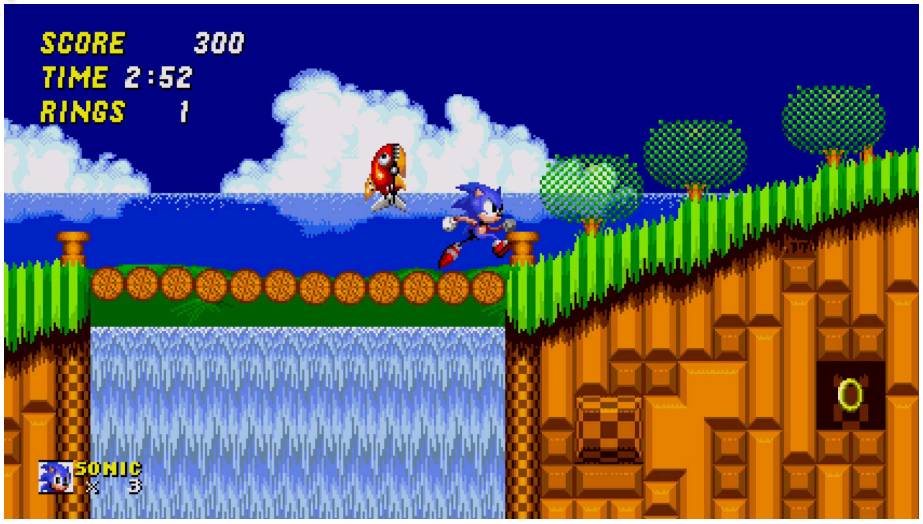
In this first exit, Sonic finds himself opposing the mechanized forces of Dr. Ivo Robotnik.
The mad scientist conquered the South Island, enclosing his population of animals in cybernetic shells as he tries to recover the legendary Emeralds of chaos.

Driving Sonic, it's up to the player to make his way through the robotic animal army, collect the emeralds for yourself, and stop Robotnik's schemes once and for all.

However, Sonic was not and still is not a simple catch-and-drop platform game - be careful but it is the first example of a running platform.

Most stages of the game are built to show the character's speed and are full of springs, slopes and loops to send Sonic at ever higher speeds. Along the
road you will collect scattered rings that will act as “vital energy”. As long as you're holding at least one ring, a collision with an enemy or an





environmental hazard won't kill our hero.

There are also other things to grab: protective shields, temporary invincibility. Even ultra-fast running shoes that bring Sonic down at an even more relentless pace. Each world is divided into three acts at the end of which the evil mad scientist and one of his death machines will be confronted.

It's not an excessively large number of stages, but the game as a whole is highly playable - even after 29 years.

How did it age? All right, then.

The graphics are still a joy for the eyes in both the 16bit and 8bit versions especially when played on an old CRT TV .

Few people realize how difficult it was to create a graphics engine like the one in this game, which allowed for incredible speed and a wide variety of animations, objects, sprites and parallax effects. The result was wonderful in '91 and still looks great today.

Verdict

Sonic the Hedgehog is a wonderful classic game that must be tested by the new generations and reworked by us with the same passion as 1991 .
To conclude, quoting the beautiful version for Master System, the programmers have completely rewritten the game and therefore the version for the small 8bit of Sega is another game. Unbelievable!

This review was made with tests on real hardware, but I recommend you try it in every possible way. Emulation, Collection on console (Ps2, Xbox, Wii) and even on Gameboy advance.

How can I answer the initial question??? Yeah, it's still fun to run as Sonic, even after 29 years.

by **Carlo N. Del Mar Pirazzini**

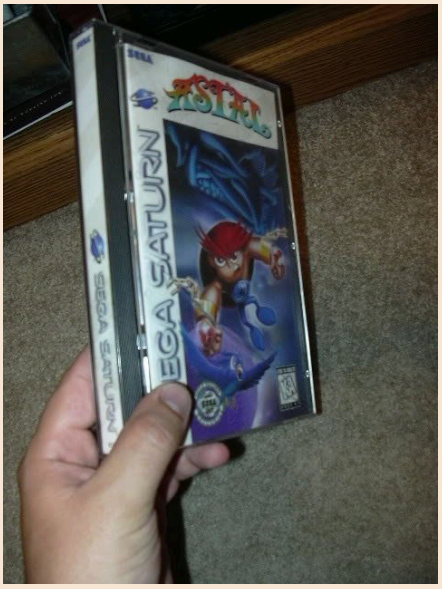
OUR FINAL SCORE

» Gameplay 95%
Fast, very fast, hyperkinetic. Everything is done using the directional cross and a button. Simple and intuitive.

» Longevity 99%
Each level has secret passageways, hidden objects, almost unreachable platforms. Exploring it all after 29 years is really hard. Give it a shot.

ASTAL

Editor: Sega
Developer: Sega
Platform: Sega Saturn
Genre: Platform



"Astal, the best platform... no one's ever played!"

Yes, because when it was developed, with its magnificent 2D at maximum levels, with its colorful, complex and plenty of sprites on the screen, the world of consoles was in the midst of the 32-bit war that was fought at the sound of 3D games. We all wanted to play with Super Mario 64, Tomb Raider or Resident Evil! 3D, all titles that had taken the video game industry by storm.

SEGA stood at the corner with his Saturn console trying to recover with 3D games not all perfectly successful, but making small graphic jewels like Astal, which was surprisingly snubbed by players and critics.

Astal is a fairly simple 2D platform game, no doubt. But what stands out is its amazing graphics, beautiful soundtrack, fun gameplay and an almost interesting story.

The saving of the world by Astal, initially a gruesome and irascible hero who will have to save his world and his beauty from the conquest plans of a demon and his servant.

A simple story, seen so many times (well, why not?! E/N) in video games, and that's exactly how Astal begins, one of the first games that really showed the ability to use 2d graphics on Saturn.

The protagonist moves across 16 levels, interacting with the surrounding environment in the most classic way of platforms. It can strike enemies by grabbing and throwing them, or by hitting them with a devastating "head butt" or with a double punch (Bud Spencer style). It can also suck and blow away enemies with a strong gust of wind.

In the game we will be helped by a kind of crow who will help us in battle with some special attacks very useful in some situations.

Graphics alone is worth the gaming experience. The world of play with its levels is beautifully realized wonderful.

Super detailed backgrounds, fluid animations, beautifully designed character sprites with beautifully animated end-of-level Boss.
The soundtrack is as beautiful as the graphics. Each level has its own unique theme, from symphonic melody from

the epic sound of level 1, to relaxing melody and new age in forest levels. The soundtrack of the game uses real instruments, giving it a symphonic feel.

There is also a hidden Japanese pop song that has been removed from the game in the United States version that you can access if you put the game CD in your stereo.

OUR FINAL SCORE

» Gameplay 90%

It is well structured, the character is well characterized and drives with extreme simplicity. The controls are easy and you take your hand right away. The difficulty is never too great...

» Longevity 65%

... and that's why it ends too easily. The sixteen levels are beautiful but too simple and structured in a single way. This flaw compromises the level of challenge, especially for those who are familiar with this kind of games.

The only flaw I can find with music is that sometimes it seems to prevail over the dialogue expressed during the intermediate sequences and in the game itself, so sometimes it can be difficult to hear what the characters say, but maybe it's good in a way. .

The dubbing of the game certainly leaves room for improvement. The storyteller who moves the story has a soft and reassuring tone appropriate for his voice, but often lacks emotion and seems to read the dialogues from the newspaper. Besides, I don't like Astal's voice. She looks like a grown woman who tries too hard to imitate the voice of a grumpy little boy. The demon, Jerado, has an impressive tone and is conducive to his voice, and is probably the best of the actors.
Let's get to the painful parts. There's a slight flaw in gameplay. The game is terribly linear. The level only crosses so, there are no hidden paths, secret levels or bonus rooms.

Despite the gorgeous graphics and high production values, Astal falters slightly due to one of the main factors: its ease of play. Although the game boasts 16 levels, these are fairly short and some of them are dedicated only to battles against bosses. The difficulty is never exaggerated and never punitive.

Adding up is a fun and beautiful game to watch. A hidden gem of the Saw Saturn that thisyear marks its 25th birthday since its release in the West.

A game that honors the beautiful 2d platforms but that will perhaps be too easy for the most "sloppy" and that unfortunately has not had a series of followers and future developments.

If you evaluate graphics alone it would be an absolute masterpiece, but not only of beauty you can live...

by **Carlo N. Del Mar Pirazzini**

POWER SPIKES

Developer: V-System/Data East
Year: 1989
Platform: Arcade
Genre: Sport simulation

Between the late 1980s and the first half of the 1990s, Japanese software house Video System created a trilogy dedicated to volleyball. If Super Volleyball was very successful in 1989, Power Spikes – and we can say it out loud – can rightly be considered its very valuable natural heir.
From its illustrious predecessor, in fact, Power Spikes takes full inspiration: the unusual - for the type of sport – scrolling side view and, basically, the same mechanics. Furthermore it adds a much more refined gameplay thanks to a much more "realistic" physics.

This sequel is definitely interesting since it improves everything that caused so much excitement in the prequel and this is what has always remained in the hearts of many fans. Even if perhaps, at least for the writer of this article, it is slightly lacking in charm. However, this is not a fault: after all, you never forget your first love and Super Volleyball had the merit of making many videogame players aware of this wonderful sport that in Italy has always had a great following. In particular, the 1990s saw its greatest splendor since - among the many international trophies won by men's national team – Italy brought home three consecutive World Cups during '90,' 94 and '98 editions.

Power Spikes offers additional contents. You can choose between men's and women's national teams. It's fair to remember how the different versions of the title, Asian and Western, allowed you to select different teams. The Western edition let you choose the national teams from Italy, France, Germany, the United Kingdom, Spain, Sweden, Norway and Brazil, but only from men's teams. The women's teams were only included in the different editions distributed in Asia (oddly, Italians gamers remember this feature very well), namely for Japan, South Korea and China's crews.

As always, the goal is to win all games to climb to the top step of the podium wearing the coveted gold medal and, as usual, the title allows us to play the final stages of the fifth set. Six matches for glory to be won again and again, as the final will always start with a significant 8-12 gap for the CPU. It's the same mechanic seen in Super Volleyball, but with different teams to face.
The strong point of Power Spikes is certainly its gameplay that is much more refined despite sporting a much simplified set of commands. The title uses a button to execute all fundamentals ("blocks" included) while in Super Volleyball this was entrusted to a second button.

The hits to be carried out essentially remain those of the predecessor (after all, we are always talking about volleyball, ed.), but they are greatly improved by a more credible physics. Let's explain it: the impact of the ball with the players has many more facets and the reception, seemingly identical, can have different results such as generating an unsuitable trajectory for the lifter who is forced to use the underarm pass (instead of a lift) hampering the effectiveness of those who have to attack. It also allows for rocky actions that, however, increase the variety of gameplay. Although the game offers a general indication (by means of an arrow) on where

the ball could inpact, the difficulty is insidious because the ball, as said, can always take a different trajectory. Often and willingly even a slight movement of our volleyball player can result in a clean or dirty play and it is not said that the latter, especially in attack, is not a winner.

It is precisely this that makes the difference with the past: the possibility of making the same hits and throws in an absolutely different way and with different results. A widening of the range of action that is not indifferent and also increases satisfaction in terms of gameplay. To give you a better idea, this evolution of gameplay is comparable to the transition experienced by Kick Off users who played Kick Off 2 and, later, Goal.

In a game like Power Spikes, where physics affects a lot, the timing, as we said, determines a large percentage of success of receiving, passing and finding an offensive solution. The title of Video System gives a lot of importance to the fundamental of the defense allowing saves (even diving forward to recover, for example, damped balls) and back to “climb in the air” to intercepts treacherous trajectories, but also the attack has its part and the success of the shots depends on when you press the button at the right time (when the ball turns red). Again, we can attack the way we want. Although – if you are quite unscrupulous – squeeze from the second line. Often and gladly it can also be a real wild card. You can make blocks, strike “poisonous” lobs, fake attacks and so on. But you can also make a lot of mistakes. In Power Spikes, many times, the one who makes the least mistake will win, because the mistake is always around the corner even in seemingly harmless balls. And a wayward or sloppy rise can ruin an otherwise perfect receptions. Never take anything for granted.

As a tradition there are also services with special strokes: Tower, Phantom and Towing Service.

From a technical point of view, the game has basically the same impact as the illustrious predecessor but with less bright colors in favor of more animations. The latter feature is evident throughout the game, starting with the audience that is on the move during the actions. The side view is fluid as players move with multiple frames and variety. It should be noted that the action of the gameplay basically has no smudges. There are also the usual portraits of the players of our sextet, both in the presentation phase and after the points grounded. And, if you choose Italy, some players on the field refer to real athletes such as Andrea Lucchetta and Luca Cantagalli recognizable by their haircut and features.

The sound, albeit very much in theme, and in "ball”, is not as memorable as the scores of Super Volleyball. This does not detract from its excellent workmanship. And the comments at the end of each winning attack such as “Nice Spike”, “Nice Block” are always present and represent a trademark of the series.
Power Spikes is a darn good game, probably the best in the Video System trilogy, even though it's not as "charming” as the forefather. The graphic improvements made, although the colors are not as warm, the very beautiful animations and a colossal extension of the gameplay that offers many more unknowns and different situations at each game, let this game to be remembered as one of the best arcades dedicated to volleyball ever.

by **Edoardo Ullo**

OUR FINAL SCORE

» Gameplay 93%

Clear improvement in gameplay thanks to enhanced and credible applied physics. The variety of offensive solutions and defensive strikes has been expanded. Greater difficulty but more satisfactory gameplay.

» Longevity 90%

Compared to the predecessor we find more teams to choose from and, in Asian versions, also the women's teams.

CHUBBY GRISTLE

Year: 1988
Developer: Teque Software Development Ltd.
Publisher: Grandslam Interactive
Version: Commodore 64
Genre: Platform

We are slowly heading towards the end of the pandemic and spring begins to feels like it's always been, that is, rainy and crazy weather.
Many of us, as always, take advantage of staying at home to dust off games that we never ended. I know I'm being a little repetitive but I'll tell you how it went during my period of forced confinement and how it's still going until the first taste of the freedom we need after almost three months of lockdown.

So on a rainy night in Milan with thunders, lightnings and flooding (in the hope of avoiding a black-out) I searched my game library for a small Commodore 64 game, one of the many that at the time did not obtain the lights of success: Chubby Gristle!

Well, the title name itself may not be very inviting, indeed it actually sounds like the title of a comedy movie. And in fact it's a little bit like a fun movie, just look at the funny protagonist, a frankly overweight man who wanders around many different levels and screens in search of his beloved food.

I didn't know that at the time when I first had a look at it. Only some time later did I find out that the name of the game would mean a rather chubby person.

Basically, the aim of this weird game is to cross the all the screens in three different settings: a supermarket, a pub and finally our beautiful salon furnished with all the treats, i.e. the protagonist's favorite food.

The gameplay follows the scheme of a classic platform game, where you have to raid and collect all the scattered objects on every screen, avoiding enemies with pinpoint precision. Watch out for jumps since even falling from a minimum height loses one of the three lives available.

Needless to say, to keep us company and motivate us to continue the game despite the high difficulty (by the way, which game was easy to finish at the time?) there is a nice background music that you will hardly be able to cancel out of your head.

In order to continue and get to the end you will need many hours of "training" (which the protagonist obviously does not!). It is quite hard to move without falling down, to avoid enemies in the hardest spots of each level and collecting all the objects and treats. So be very careful!

Some items should not actually be collected, otherwise they will be harmful later, compromising your entire game! Last but not least, the graphics: they look like those ZX Spectrum platform games a lot, because of some typical monochrome areas. The game was indeed published on several home computers, including the aforementioned 8-bit Sinclair, so the graphics did not differ much from the C64 version. However, it is always worth trying it at least once and, why not?, re-evaluate it.

I recall that this game was even on sale in my trusted computer store back in the 80s. It was shown among the headline titles who made the story of the C64, although I never bought it.

So, we have come to the end (hopefully) of the pandemic and we are getting ready to go out, see our friends and play this fun game together!

I wish you all a successful return to normal life and hope that you will emerge victorious from both the challenges!

by **Daniele Brahimi**

OUR FINAL SCORE

» Gameplay 70%
You die easily in this game and you 'll need a lot of practice.

» Longevity 80%
Not very long, but surely challenging.

Synergy means behavior of integral, aggregate, whole systems unpredicted by behaviors of any of their components or subassemblies of their components taken separately from the whole.

Buckminster Fuller, inventor, designer, architect

The generic definition of **synergy** is: the relationship between elements or forces working towards the same scope. Simple, isn't it? Here is the key! When different elements work together to achieve a common purpose, the results are extraordinary.

I've borrowed Buckminster Fuller's phrase in the title to talk again about a topic that has always been dear to me: the collaboration between us enthusiasts to achieve always a better result.

In all fields of human work we have had this demonstration, but to bring it back to our world, just think about opensource, the projects shared on GitHub or, to remind a couple of well known stories, those companies born from a communion of intentions of few people who shared the same dream!

We, at RetroMagazine World, have always made collaboration our flag. None of us, however prepared or technically gifted, would be able to carry on such a complex magazine and all the projects which compose the world of RMW, without being overwhelmed. That is why there are so many of us and why we have always invited all of you to participate and collaborate.

An example that I'd like to quote is the article **Antonino Porcino** sent us for the Italian version of the fanzine; it included the full listing of a text adventure first published in the 80s by a popular magazine. After reading it I felt the duty to give my contribution and, as soon as we talked about it in the editorial team, **Ermanno** also provided his contribute. We could not add this article in the English version because the game was entirely in Italian; sorry... :-(
Another example that, thanks to the friendship that binds me to **Marco**, I can follow closely, is the development of the **PlayEdicola** tool, a front-end software to load and play retrogames. Born as a personal project, thanks to the contribution of other volunteers, it has grown significantly and it will soon be available to everyone. And not to mention the game **Leonardo** is developing together with a Dutch coder...

The barrier of individualism has been broken again... To boldly go where no man has gone before...! (Ah but that's another story...).
I would like to close by anticipating that in the upcoming months we will be launching a series of projects that will require your cooperation. I will not reveal anything else, because it is still early and there is already a lot to do, as you have read in **David**'s editorial. Please continue to follow us and, if you wish and some time to spare, collaborate with us. Synergically!

Francesco Fiorentini

Disclaimer

RetroMagazine World as an aperiodic magazine entirely ad-free is a non-profit project and falls off any commercial circuit.

All the published material is produced by the respective authors and published thanks to their authorization.

RetroMagazine World is licensed under the terms of:
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)
https://creativecommons.org/licenses/by-nc-sa/4.0/

This is a human-readable summary of (and not a substitute for) the license.

You are free to:
Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material
The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:
Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

RetroMagazine World
Year 1 - Issue 1 - JUNE 2020

Chief Editor
Francesco Fiorentini
Managing Editor
David La Monaca
Editing Manager
Marco Pistorio
Web Manager
Giorgio Balestrieri